

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision
of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-337864

(P2001-337864A)

(43) 公開日 平成13年12月7日 (2001.12.7)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード [*] (参考)
G 0 6 F 12/14	3 1 0	G 0 6 F 12/14	3 1 0 K 5 B 0 1 7
1/00	3 7 0	1/00	3 7 0 E 5 B 0 7 6
		12/00	5 3 7 M 5 B 0 8 2
12/00	5 3 7	15/00	3 3 0 A 5 B 0 8 5
15/00	3 3 0	9/06	6 6 0 E

審査請求 未請求 請求項の数38 O L (全 31 頁)

(21) 出願番号 特願2000-300561 (P2000-300561)

(22) 出願日 平成12年9月28日 (2000.9.28)

(31) 優先権主張番号 特願2000-84706 (P2000-84706)

(32) 優先日 平成12年3月22日 (2000.3.22)

(33) 優先権主張国 日本 (J P)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 荒井 正人

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72) 発明者 松本 日高

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(74) 代理人 100075096

弁理士 作田 康夫

最終頁に続く

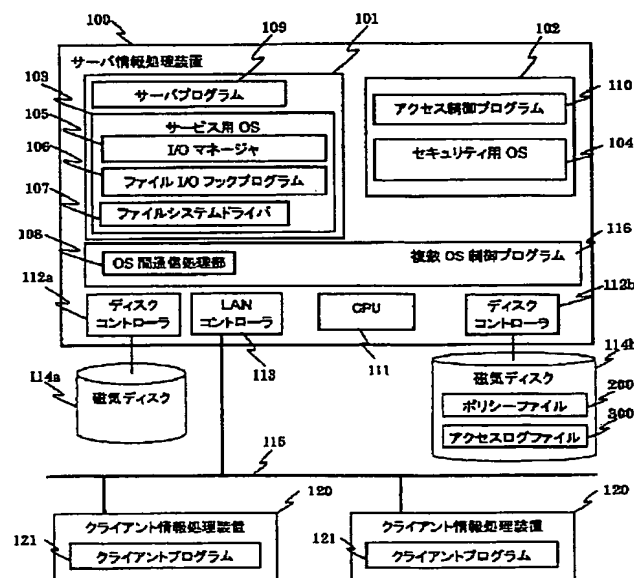
(54) 【発明の名称】 アクセス制御システム

(57) 【要約】

【課題】 ネットワークからの侵入者がいかなるユーザー権限を利用して不正なファイル読み出しや書き込みを試みても、該アクセスの抑止が可能なアクセス制御システム及びその方法を提供する。

【解決手段】 特定のファイルへのアクセスを、特定のユーザーが特定のプログラムを使用した場合に限り許可するといったポリシーを用いる。さらに、ポリシーをポリシーファイル200に登録し、ファイルI/Oフックプログラム106がフックしたアクセス情報を、OS間通信処理部108を介してセキュリティ用OS104上のアクセス制御プログラム110に渡し、前記ポリシーに基づいたアクセス制御を行う。

図1



100: サーバ情報処理装置

101: サービス用OSが管理するメモリ

102: セキュリティ用OSが管理するメモリ

120: クライアント情報処理装置

【特許請求の範囲】

【請求項1】各種処理に必要な情報を格納するための記憶部と、

記憶媒体に格納されたファイルへの入出力処理部と、
前記ファイルへのアクセス制御ポリシーを記述した、前記記憶部中のポリシーファイルと、

当該ファイルアクセス要求の正当性を前記アクセス制御ポリシーに基づき判定するアクセス制御処理部と、

前記入出力処理部を用いたファイルアクセス要求の発行を監視し、発行されたファイルアクセス要求を前記アクセス制御処理部に伝え、正当性判定結果を前記アクセス制御処理部から受信する監視処理部と、

前記入出力処理部とアクセス制御処理部と監視処理部各々が使用する前記記憶部中の記憶領域と、前記ポリシーファイルとを、前記前記入出力処理部とアクセス制御処理部と監視処理部のいずれとも異なるアクセス実行処理部から保護するための排他制御部と、を備え、

前記ポリシーファイルは、アクセス制御ポリシーとして、アクセス対象となるファイルについて、アクセス要求発行元とアクセス実行処理部とアクセスタイプとを特定する情報を備え、

前記監視処理部は、アクセス要求発行元とアクセス実行処理部とアクセスタイプとを特定する情報を用いて、前記発行されたファイルアクセス要求を伝えるアクセス制御システム。

【請求項2】請求項1記載のアクセス制御システムであって、

前記アクセス制御ポリシーは、前記ファイルに対して禁止されたアクセスタイプと、該禁止されたアクセスが発生した場合にアクセス要求発行元に返すべきエラーコードと、例外として該アクセス要求を許可されたアクセス実行処理部とアクセス要求発行元とを特定する情報とからなるアクセス制御システム。

【請求項3】請求項2記載のアクセス制御システムであって、

前記アクセス制御ポリシーに記すアクセス実行処理部は、プログラムであり、該プログラムのパス名と、該プログラムの特徴値との組合せで特定するアクセス制御システム。

【請求項4】請求項3記載のアクセス制御システムにおいて、

さらに、ファイルアクセス要求内容を登録するアクセスログファイルを備え、前記アクセス制御処理部は、前記ファイルアクセス要求を前記ポリシーファイルの記述と照合し、正当性の判定結果を前記監視処理部に送信するとともに、

前記アクセス要求が許可されるものであれば、当該アクセス実行処理部の特徴値を、前記監視処理部に送信し、前記アクセス要求が前記アクセス制御ポリシーに違反するものであれば、該ファイルアクセス要求内容を、前記

アクセスログファイルに登録するアクセス制御システム。

【請求項5】請求項4記載のアクセス制御システムにおいて、

さらにオープンファイルテーブルを備え、

前記アクセス制御処理部により正当であると判定された前記ファイルアクセス要求のアクセスタイプがオープンアクセスである場合、

前記アクセス制御処理部からレスポンス情報として取得した、アクセスタイプと、前記アクセス対象となるファイルと、前記アクセス要求発行元と、アクセス実行処理部とを特定する情報を、前記オープンファイルテーブルに登録する処理部と、

前記アクセス要求としてリードアクセスまたはライトアクセスが発行された場合は、

前記オープンファイルテーブルを検索し、前記アクセス要求の正当性を判定する処理部を備えるアクセス制御システム。

【請求項6】請求項5記載のアクセス制御システムであって、

前記監視処理部は、さらに、

前記オープンファイルテーブルに登録されていないリードアクセス要求またはライトアクセス要求を検知した場合は、前記アクセス制御処理部を介して当該アクセス要求内容を前記アクセスログファイルに登録する処理部を備えるアクセス制御システム。

【請求項7】請求項6記載のアクセス制御システムであって、

前記監視処理部は、さらに、

ファイルクローズ要求を検知した場合には前記オープンファイルテーブルから該当する情報を削除する処理部を備えるアクセス制御システム。

【請求項8】請求項4記載のアクセス制御システムにおいて、

前記アクセス制御処理部は、

当該属性情報と前記ポリシーファイルの記述との前記照合を、前記ファイルアクセス要求のアクセスタイプがオープンアクセスであり、且つ当該ファイルアクセス要求の属性情報として、リードアクセスまたはライトアクセスに関する情報が含まれる場合におこなうアクセス制御システム。

【請求項9】請求項8記載のアクセス制御システムであって、

前記監視処理部は、さらに、

前記ファイルアクセス要求が正当であると判定された場合、前記前記監視処理部は前記アクセス実行処理部の特徴値を算出し、前記アクセス制御処理部から受信した前記特徴値とを比較する処理部と、

一致した場合には前記アクセス要求を許可する処理部と、

一致しない場合には前記ファイルアクセス要求を無効にすると共に、前記アクセス制御処理部を介して当該ファイルアクセス内容を前記アクセスログファイルに登録する処理部とを備えるアクセス制御システム。

【請求項 10】ファイルへの入出力処理部と排他的に使用する第 1 の記憶処理部を管理する第 1 の OS と、排他的に使用する第 2 の記憶処理部を管理する第 2 の OS と、前記第 1 の OS と第 2 の OS との間でデータ通信するための通信処理部とを具備した情報処理システムにおいて、

前記第 1 の OS は、管理する前記ファイル入出力処理部へ発行されたファイルアクセス要求を監視する監視処理部を備え、前記第 2 の OS は、前記ファイルアクセス要求の正当性をアクセス制御ポリシーに基づき判定するアクセス制御処理部を備え、前記監視処理部は、前記アクセス要求を、前記通信処理部を介して、前記アクセス制御処理部に伝え、正当性判定結果を、前記通信処理部を介して、前記アクセス制御処理部から受信する情報処理システム。

【請求項 11】請求項 10 記載の情報処理システムにおいて、

前記第 2 の OS は、前記アクセス制御ポリシーとして、アクセス対象となるファイルについて、アクセス要求発行元とアクセス実行処理部とアクセスタイプとを特定する情報を記すポリシーファイルを備える情報処理システム。

【請求項 12】請求項 11 記載の情報処理システムであって、

前記アクセス制御ポリシーに記すアクセス実行処理部は、プログラムであり、該プログラムのパス名と、該プログラムの特徴値との組合せで特定する情報処理システム。

【請求項 13】請求項 12 記載の情報処理システムにおいて、

前記第 2 の OS は、さらに、ファイルアクセス要求内容を登録するアクセスログファイルを備え、

前記アクセス制御処理部は、

前記ファイルアクセス要求を前記ポリシーファイルの記述と照合し、正当性の判定結果を前記監視処理部に送信するとともに、

前記アクセス要求が許可されるものであれば、当該アクセス実行処理部の特徴値を、前記監視処理部に送信し、前記アクセス要求が前記アクセス制御ポリシーに違反するものであれば、該ファイルアクセス要求内容を、前記アクセスログファイルに登録する情報処理システム。

【請求項 14】請求項 13 記載の情報処理システムにおいて、

前記第 1 の OS は、さらにオープンファイルテーブルを備え、

前記アクセス制御処理部により正当であると判定された

前記ファイルアクセス要求のアクセスタイプがオープンアクセスである場合、

前記アクセス制御処理部からレスポンス情報として取得した、アクセスタイプと、前記アクセス対象となるファイルと、前記アクセス要求発行元と、アクセス実行処理部とを特定する情報を、前記オープンファイルテーブルに登録する処理部と、

前記アクセス要求としてリードアクセスまたはライトアクセスが発行された場合は、

前記オープンファイルテーブルを検索し、前記アクセス要求の正当性を判定する処理部を備える情報処理システム。

【請求項 15】請求項 14 記載の情報処理システムであって、

前記監視処理部は、さらに、

前記オープンファイルテーブルに登録されていないリードアクセス要求またはライトアクセス要求を検知した場合は、前記アクセス制御処理部を介して当該アクセス要求内容を前記アクセスログファイルに登録する処理部を備える情報処理システム。

【請求項 16】請求項 15 記載の情報処理システムであって、

前記監視処理部は、さらに、

ファイルクローズ要求を検知した場合には前記オープンファイルテーブルから該当する情報を削除する処理部を備える情報処理システム。

【請求項 17】請求項 13 記載の情報処理システムであって、

前記アクセス制御処理部は、当該属性情報と前記ポリシーファイルの記述との前記照合を、前記ファイルアクセス要求のアクセスタイプがオープンアクセスであり、且つ当該ファイルアクセス要求の属性情報として、リードアクセスまたはライトアクセスに関する情報が含まれる場合に行う情報処理システム。

【請求項 18】請求項 17 記載の情報処理システムであって、

前記監視処理部は、さらに、

前記ファイルアクセス要求が正当であると判定された場合、前記前記監視処理部は前記アクセス実行処理部の特徴値を算出し、前記アクセス制御処理部から受信した前記特徴値とを比較する処理部と、

一致した場合には前記アクセス要求を許可する処理部と、

一致しない場合には前記ファイルアクセス要求を無効にすると共に、前記アクセス制御処理部を介して当該ファイルアクセス内容を前記アクセスログファイルに登録する処理部とを備える情報処理システム。

【請求項 19】ファイルへの入出力処理部と排他的に使用する第 1 の記憶処理部を管理する第 1 の OS と、排他的に使用する第 2 の記憶処理部を管理する第 2 の OS

と、前記第1のOSと第2のOSとの間でデータ通信するための通信処理部とを具備した情報処理システムに用いるアクセス制御システムであって、

前記第1のOSに組み込み、管理する前記ファイル入出力処理部へ発行されたファイルアクセス要求を監視する監視処理部と、前記第2のOSが制御し、前記ファイルアクセス要求の正当性をアクセス制御ポリシーに基づき判定するアクセス制御処理部と、を備え、

前記監視処理部は、前記アクセス要求を、前記通信処理部を介して、前記アクセス制御処理部に伝え、正当性判定結果を、前記通信処理部を介して、前記アクセス制御処理部から受信する情報処理システム。

【請求項20】ファイルへの入出力処理部と、前記ファイルを格納する記憶処理部とを備えた情報処理装置に読み込まれ、実行され、前記情報処理装置上にアクセス制御システムを構成させるプログラムと前記プログラムが使用するファイルとを格納した記憶媒体であって、前記プログラムが使用するファイルは、アクセス制御ポリシーを記述したポリシーファイルであって、前記アクセス制御ポリシーは、アクセス対象となるファイルについて、アクセス要求発行元とアクセス実行処理部とアクセスタイプとを特定する情報を記したものであり、

前記プログラムは、アクセス制御プログラムと、監視プログラムとを備え、前記アクセス制御プログラムは、前記情報処理装置に、

当該ファイルアクセス要求の正当性を前記アクセス制御ポリシーに基づき判定させ、

前記監視プログラムは、前記情報処理装置に、

前記入出力処理部を用いたファイルアクセス要求の発行を監視させ、

発行されたファイルアクセス要求を、アクセス要求発行元とアクセス実行処理部とアクセスタイプとを特定する情報をを用いて、前記アクセス制御プログラムに伝えさせることを特徴とする記憶媒体。

【請求項21】各種処理に必要な情報を格納するための記憶部と、サービス提供の役割を有する第1のプログラムと前記第1のプログラムのサービスを利用する第2のプログラムとが相互に通信するためのプロセス間通信処理部と、

前記第1のプログラムと第2のプログラムとを特定する情報と、前記第1のプログラムと第2のプログラムとによるプロセス間通信に関するアクセス制御ポリシーを記述した、前記記憶部中の通信制御ポリシーファイルと、当該プロセス間通信要求の正当性を前記通信制御ポリシーに基づき判定するアクセス制御処理部と、

前記第1のプログラムから前記プロセス間通信処理部へ発行される第1の通信要求を、前記第1のプログラムを特定する情報をを用いて前記アクセス制御処理部に伝え、前記第1の通信要求の正当性判定結果を前記ア

クセス制御処理部から受信する第1の監視処理部と、前記第2のプログラムから前記プロセス間通信処理部へ発行される第2の通信要求を、前記第1のプログラムを特定する情報をを用いて前記アクセス制御処理部に伝え、前記第2の正当性判定結果を前記アクセス制御処理部から受信する第2の監視処理部と、

前記プロセス間通信処理部と前記アクセス制御処理部と前記第1の監視処理部と前記第2の監視処理部が各々利用する記憶領域と、および前記通信制御ポリシーファイルを、前記プロセス間通信処理部とアクセス制御処理部と第1の監視処理部と第2の監視処理部のいずれにも該当しないアクセス実行処理部から保護するための排他制御部と、を備えるアクセス制御システム。

【請求項22】請求項21記載のアクセス制御システムであって、

前記第1のプログラムを特定する情報は、前記第1のプログラムを一意に識別可能な情報と、サービスの種類を識別するためのサービス番号との組合せであり、

前記第2のプログラムを特定する情報は、前記第2のプログラムを一意に識別可能な情報と、該プログラムの利用者情報と、特徴値との組合せであるアクセス制御システム。

【請求項23】請求項22記載のアクセス制御システムにおいて、

さらに、プロセス間通信処理を実行中のプログラムの情報を登録する通信管理テーブルを備え、

前記アクセス制御処理部は、

現在プロセス間通信可能な状態にある前記第1のプログラムを特定する情報と、該第1のプログラムとプロセス間通信処理中の前記第2のプログラムを特定する情報とを、前記通信管理テーブルに登録して保持し、

前記第2の監視処理部からプロセス間通信要求を伝えられた場合は、前記通信管理テーブルを検索し、第1のプログラムがプロセス間通信可能な状態にあり、且つ該第1のプログラムとプロセス間通信処理中の第2のプログラムが存在するか否かを判定することを特徴とするアクセス制御システム。

【請求項24】請求項23記載のアクセス制御システムにおいて、

前記アクセス制御処理部は、現在プロセス間通信可能な状態にある第1のプログラムを特定する情報として、該第1のプログラムを一意に識別可能な情報と、サービスの種類を識別するためのサービス番号と、該第1のプログラムが指定した暗証情報とを前記通信管理テーブルに登録し、

該第1のプログラムとプロセス間通信処理中の第2のプログラムを特定する情報として、該第2のプログラムを一意に識別可能な情報を前記通信管理テーブルに登録して保持することを特徴とするアクセス制御システム。

【請求項25】請求項24記載のアクセス制御システム

において、
さらに、プロセス間通信要求内容を登録する通信ログファイルを備え、
前記アクセス制御処理部は、
前記第1の監視処理部から伝えられる、前記第1のプログラムによるプロセス間通信要求を前記通信制御ポリシーファイルの記述と照合し、該プロセス間通信要求が前記通信制御ポリシーに登録されていれば、該第1のプログラムを特定する情報を、前記通信管理テーブルに登録する処理部と、
前記第1のプログラムによる該プロセス間通信要求が前記通信制御ポリシーに登録されていなければ、該プロセス間通信要求内容を、前記通信ログファイルに登録する処理部と、
前記第2の監視処理部から伝えられる、前記第2のプログラムによるプロセス間通信要求を前記通信制御ポリシーファイルの記述と照合し、該プロセス間通信要求が前記通信制御ポリシーにて許可されており、かつ前記第1のプログラムと通信可能な状態であれば、該第2のプログラムを特定する情報を前記通信管理テーブルに登録して、前記暗証情報を通信管理テーブルから取り出して前記第2の監視処理部に返信する処理部と、
前記第2のプログラムによる該プロセス間通信要求が前記通信制御ポリシーに登録されていなければ、該プロセス間通信要求内容を、前記通信ログファイルに登録する処理部と、を備えることを特徴とするアクセス制御システム。
【請求項26】請求項25記載のアクセス制御システムであって、
前記アクセス制御処理部は、さらに、
前記第1の監視処理部または第2の監視処理部からプロセス間通信の終了要求を伝えられた場合には、前記通信管理テーブルから該当する情報を削除する処理部を備えるアクセス制御システム。
【請求項27】請求項26記載のアクセス制御システムであって、
前記第1の監視処理部は、
前記第1のプログラムからプロセス間通信要求が発行された場合に、該第1のプログラムを特定する情報を取得してから、前記発行されたプロセス間通信要求を前記アクセス制御処理部に伝えるアクセス制御システム。
【請求項28】請求項27記載のアクセス制御システムであって、
前記第2の監視処理部は、
前記第2のプログラムからプロセス間通信要求が発行された場合に、該第2のプログラムを特定する情報を取得してから、前記プロセス間通信の開始要求を前記アクセス制御処理部に伝えるアクセス制御システム。
【請求項29】排他的に使用する第1の記憶処理部を管理する第1のOSと、排他的に使用する第2の記憶処理

部を管理する第2のOSと、前記第1のOSと第2のOSとの間でデータ通信するための通信処理部とを具備した情報処理システムにおいて、
前記第1のOSは、前記通信処理部へ発行されたプロセス間通信要求を監視する第1の監視処理部を備え、前記第2のOSは、前記プロセス間通信要求の正当性を通信制御ポリシーに基づき判定するアクセス制御処理部を備え、前記第1の監視処理部は、前記第1のOS上で動作する第1のプログラムを特定する情報を用いて、前記発行されたプロセス間通信要求を、前記通信処理部を介して前記アクセス制御処理部に伝え、正当性判定結果を、前記通信処理部を介して、前記アクセス制御処理部から受信する情報処理システム。
【請求項30】請求項29記載の情報処理システムにおいて、
さらに、排他的に使用する第3の記憶処理部を管理する第3のOSを備え、
前記第3のOSは、通信処理部へ発行されたプロセス間通信要求を監視する第2の監視処理部を備え、
前記第2の監視処理部は、第3のOS上で動作する第2のプログラムを特定する情報を用いて、前記発行されたプロセス間通信要求を、前記通信処理部を介して、前記アクセス制御処理部に伝え、正当性判定結果を前記通信処理部を介して、前記アクセス制御処理部から受信する情報処理システム。
【請求項31】請求項30記載の情報処理システムであって、
前記第2のOSは、前記通信制御ポリシーとして、前記プロセス間通信が可能な第1のプログラムを特定する情報と、該第1のプログラムとの通信が許可された第2のプログラムを特定する情報との組合せを記すポリシーファイルを備えることを特徴とする情報処理システム。
【請求項32】請求項31記載の情報処理システムであって、
前記通信制御ポリシーに記す第1のプログラムを、該プログラムのパス名と、サービスの種類を識別するためのサービス番号との組合せで特定し、
前記通信制御ポリシーに記す第2のプログラムを、該プログラムのパス名と、該プログラムの利用者情報と、特徴値との組合せで特定することを特徴とする情報処理システム。
【請求項33】請求項32記載の情報処理システムにおいて、
前記通信処理部は、前記第1のOSと第2のOSと第3のOSとの間で行われるデータ通信を制御し、
前記第2のOSは、さらに、プロセス間通信処理を実行中のプログラムの情報を登録する通信管理テーブルを備え、
前記アクセス制御処理部は、現在プロセス間通信可能な状態にある第1のプログラムを特定する情報と、該第1

のプログラムとプロセス間通信処理中の前記第3のOS上で動作する第2のプログラムを特定する情報とを、前記通信管理テーブルに登録して保持し、

前記第2の監視処理部からプロセス間通信要求を伝えられた場合は、前記通信管理テーブルを検索し、第1のプログラムがプロセス間通信可能な状態にあり、且つ該第1のプログラムとプロセス間通信処理中の第2のプログラムが存在するか否かを判定することを特徴とする情報処理システム。

【請求項34】請求項33記載の情報処理システムにおいて、

前記アクセス制御処理部は、現在プロセス間通信可能な状態にある第1のプログラムを特定する情報として、該第1のプログラムを一意に識別可能な情報と、サービスの種類を識別するためのサービス番号と、該第1のプログラムが指定した暗証情報とを前記通信管理テーブルに登録し、

該第1のプログラムとプロセス間通信処理中の第2のプログラムを特定する情報として、該第2のプログラムを一意に識別可能な情報を前記通信管理テーブルに登録して保持することを特徴とする情報処理システム。

【請求項35】請求項34記載の情報処理システムにおいて、

前記第2のOSは、さらに、プロセス間通信要求内容を登録する通信ログファイルを備え、

前記アクセス制御処理部は、前記第1の監視処理部から伝えられる、前記第1のプログラムによるプロセス間通信要求を前記通信制御ポリシーファイルの記述と照合し、該プロセス間通信要求元が前記通信制御ポリシーに登録されていれば、該第1のプログラムを特定する情報を、前記通信管理テーブルに登録する処理部と、

該プロセス間通信要求元が前記通信制御ポリシーに登録されていない場合は、該プロセス間通信要求内容を、前記通信ログファイルに登録する処理部と、

前記第2の監視処理部から伝えられる、前記第3のプログラムによるプロセス間通信要求を前記通信制御ポリシーファイルの記述と照合し、該プロセス間通信要求元が前記通信制御ポリシーにて許可されており、かつ前記第1のプログラムと通信可能な状態であれば、該第2のプログラムを特定する情報を前記通信管理テーブルに登録して、前記暗証情報を通信管理テーブルから取り出して前記第2の監視処理部に返信する処理部と、

前記第3のプログラムによる該プロセス間通信要求元が前記通信制御ポリシーに登録されていない場合は、該プロセス間通信要求内容を、前記通信ログファイルに登録する処理部と、を備えることを特徴とする情報処理システム。

【請求項36】請求項35記載の情報処理システムであって、

前記アクセス制御処理部は、さらに、

前記第1の監視処理部または第2の監視処理部からプロセス間通信の終了要求を伝えられた場合には、前記通信管理テーブルから該当する情報を削除する処理部を備えることを特徴とする情報処理システム。

【請求項37】請求項36記載の情報処理システムであって、

前記第1の監視処理部は、

前記第1のプログラムからプロセス間通信要求が発行された場合に、該第1のプログラムを特定する情報を取得してから、前記発行されたプロセス間通信要求を前記アクセス制御処理部に伝える情報処理システム。

【請求項38】請求項37記載の情報処理システムであって、

前記第2の監視処理部は、

前記第2のプログラムからプロセス間通信要求が発行された場合に、該第2のプログラムを特定する情報を取得してから、前記プロセス間通信の開始要求を前記アクセス制御処理部に伝え、該アクセス制御処理部にて許可される場合には、前記プロセス間通信の開始を許可して、前記第1のプログラムに対して前記暗証情報を伝える処理部と、前記アクセス制御処理部にて許可されない場合には前記プロセス間通信の開始を無効にする処理部とを備えることを特徴とする情報処理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、情報処理装置が管理する情報を不正なアクセスから保護する場合に好適なアクセス制御技術に関する。

【0002】

【従来の技術】一般のコンピュータシステムでは、マルチユーザー・マルチタスクOSが備えるユーザー認証機構と、該認証結果に基づいたファイルアクセス制御機構を用いて機密情報ファイルの保護を実現しているケースが多い。具体的には、前記OSが実装された情報処理装置を利用する際に、ユーザーは必ず自己のユーザーIDとパスワードを入力し、認証を受ける。前記情報処理装置が管理する全てのファイル各々には、ファイルリードやライト等のアクセスタイプ毎に、ユーザーIDとグループIDを用いてアクセス可能なユーザーを定義したアクセスコントロールリストがセキュリティ属性情報として割り当てられている。ユーザーがアプリケーションプログラムを介してファイルへアクセスした場合、上記OSは、アクセス要求元となるユーザーのID及び該ユーザーが所属するグループIDを、アクセス対象となるファイルに割り当てられたアクセスコントロールリストと照合し、当該リストに前記ユーザーが含まれている場合に限りアクセスを許可するといった制御を行なう。

【0003】一方で、インターネットを介した情報発信、情報収集や、各種サービス提供を行なう手段としてWorldWideWeb(WWW)のサービスが広く

使われている。WWWでは、HyperText Transfer Protocol (HTTP) と呼ばれる通信プロトコルが、リクエストデータとレスポンスデータの転送に用いられている。また、WWWシステムではコンテンツの不正な差し替えや、ネットワークを経由した機密情報流出を未然に防ぐためのセキュリティ技術がいくつか用意されている。

【0004】HTTPが有するセキュリティ機構として、基本認証(Basic Authentication) と呼ばれるものがある。基本認証では、認証情報としてユーザーIDとパスワードを予めWWWサーバに登録しておき、ブラウザを通じてユーザーから送信されるユーザーIDとパスワードを、前記認証情報と比較して認証を行なう。各コンテンツへのアクセス権限設定を記述したポリシーファイルと、該ポリシーに基づくアクセス制御機構も前記WWWサーバに実装されている。同様な機構を、Common Gateway Interface (CGI) プログラムに実装し、ユーザー認証とコンテンツへのアクセス制御を実現することも可能である。

【0005】WWWでは、他のセキュリティ技術として、認証局から発行される証明書に基づき、ユーザーとWWWサーバの相互認証と通信データの暗号化を行なうことが可能である。この技術は、消費者のクレジットカード番号がネットワーク上に流れる一部のインターネットショッピングのようなサービスにおいて、必須の技術とされている。これらの技術によるユーザー(クライアント)認証は、WWWサーバ側のOSが具備するユーザー認証機構とは通常独立したものであるが、前記認証に用いるユーザーIDや、ユーザーの証明書を、OSが管理するユーザーアカウントに関連付けて使用することができるWWWサーバプログラムもある。つまり、認証されたユーザーは、OSのアクセス制御下でWWWコンテンツにアクセスする。これらのセキュリティ技術については、“Web Security: A Step-by-Step Reference Guide” Lincoln D. Stein著、Addison-Wesley Pub Co; (ISBN: 0201634899)などに記載されている。

【0006】近年増加しているインターネットサービスプログラム等に潜むセキュリティホールやバグを利用した不正アクセスを解決する他の技術としては、常駐型のファイル監視プログラムを情報処理装置上に設け、定期的にファイルの破壊の有無をチェックする方法が、特開平10-069417号公報に開示されている。これと同様の技術として、特定のファイルについてそのサイズの増減やタイムスタンプを定期的にチェックするプログラムを情報処理装置上に設け、該ファイルの変化を検知するツールが、シェアウェアやフリーウェアとして幾つか公開されている。

【0007】

【発明が解決しようとする課題】先に述べたセキュリテ

ィホールやバグに関する情報は、http://www.cert.org/等で報告され、バグ修正用のプログラムも各メーカーから配布されるようになっている。しかし、攻撃者あるいは侵入者は、あらゆる手段を用いて外部ネットワークから情報処理装置への侵入を試みる。たとえば、前記マルチユーザー・マルチタスクなOSが具備するアクセス制御機構では、アクセス要求元のユーザーIDとその所属グループに基づいてアクセスの可否を判断しているため、OSの管理者のような強い権限をもつユーザーに成りすまして侵入すれば、システム中のいかなるファイルにもアクセスできるので、システム中の情報を書き換えたり、機密情報を盗み出すことができてしまう。

【0008】また、前記ファイルの変化を定期的にチェックする手法では、ファイルが改ざんされた後の検出になるため、不正アクセスが発生したことは分かるが、それを未然に防ぐことができないと共に、ファイルの不正な読み出しについては事後検出もできない。

【0009】また、前記WWWサーバやCGIプログラムを利用してコンテンツのアクセス制御機能を実現したシステムにおいては、各コンテンツへのアクセス権限設定を記述したポリシーファイルも、その他のファイルと同じく不正アクセスの対象になる。したがって、該ポリシーファイルが破壊あるいは改ざんされた場合には、前記アクセス制御機能が正常に働かない。

【0010】本発明の目的は、アクセス要求元(サブジェクト)がいかなる権限をもつ場合でも、そのアクセス手段およびアクセス対象(オブジェクト)を制限することが可能なアクセス制御システム及びその方法を提供することにある。

【0011】本発明の他の目的は、不正なファイルアクセスを未然に防ぐことが可能なアクセス制御システム及びその方法を提供することにある。

【0012】本発明の他の目的は、アクセス権限設定を記述したポリシーファイルと、該ポリシーファイルに基づいてアクセス制御を施行するプログラムを、外部からの不正アクセスや攻撃から保護可能なアクセス制御システム及びその方法を提供することにある。

【0013】本発明の他の目的は、例えば機密情報にアクセスしながらサービスを提供するようなアプリケーションプログラムを、外部からの不正アクセスから保護可能なアクセス制御システム及びその方法を提供することにある。

【0014】

【課題を解決するための手段】上記目的を達成するために、本発明では、ファイルアクセスに関するアクセス制御ポリシーとして、より厳密にアクセス要求を特定する情報、すなわち、アクセス要求元とアクセス実行手段とアクセスタイプとを特定する情報を用いる。

【0015】さらに具体的には、特定のファイルへのア

アクセスを、特定のユーザーが特定のプログラムを用いた場合のみ許可するアクセス制御ポリシーを記述したポリシーファイルを作り、アクセス制御手段が、ファイルへのアクセスが発生したときに、前記ポリシーファイルの記述に従ってアクセスの正当性すなわち可否を判定する。前記アクセス制御ポリシーには、アクセス対象となるファイルの名称と、アクセスが許可されたユーザー名とプログラム名の組合せを、ファイルのオープン・リード・ライト・削除・リネームといったアクセスタイプ毎に予め規定しておく。

【0016】また、不正なファイルアクセスを未然に防ぐために、本発明では、前記ファイルアクセスを監視するファイルI/O（Input/Output）フック手段を前記情報処理装置に設け、該ファイルI/Oフック手段は、前記アクセスを検知した時に、当該アクセスタイプ及びアクセス対象となるファイルの名称と、該アクセスの要求元となるユーザー名及びプログラム名を取得して前記アクセス制御手段に送信する。前記アクセス制御手段は、受信内容を前記ポリシーファイルの内容と照合し、該ポリシーに違反するアクセスであれば当該アクセスを無効にし、前記ファイルI/Oフック手段を経由して前記アクセス要求元にエラーを返す。

【0017】また、前記ポリシーファイルと前記アクセス制御手段を保護するために、本発明では、1台の情報処理装置上に2つのOSと、両OS間でデータ通信するためのプロセス間通信手段と、両OSが互いに排他的に占有するメモリや磁気ディスクおよびネットワークデバイスを設ける。前記2つのOSのうち一方をアクセス監視対象となるサービス用OSとして使用し前記ファイルI/Oフック手段を前記サービス用OSのカーネルレベルのモジュールとして設ける。もう一方をセキュリティ用OSとして使用し、前記アクセス制御手段および前記ポリシーファイルを占有する磁気ディスクに格納すると共に、前記アクセス制御手段をプロセスとして動作させ、該アクセス制御手段とポリシーファイルを前記サービス用OS上の、サービスを受けるユーザが使うプロセスからアクセスできないようにする。

【0018】一方、よりセキュリティを高めるために、機密情報を取り扱うようなアプリケーションプログラム（以後、機密プログラムと称す）とのプロセス間通信に関するアクセス制御ポリシーとして、より厳密にアクセス要求を特定する情報、すなわち、アクセス要求元とアクセス実行手段とを特定する情報を用いてもよい。具体的には、特定の機密プログラムとのプロセス間通信を、特定のユーザーが特定のプログラムを用いた場合のみ許可するアクセス制御ポリシーを記述したポリシーファイルを作り、複数のアプリケーションプログラム間で通信が発生したときに、プロセス間通信手段が当該通信を検知してアクセス制御手段に通知し、アクセス制御手段が前記ポリシーファイルの記述に従って通信の正当性すな

わち可否を判定する。前記アクセス制御ポリシーには、通信機能を備える機密プログラムの名称と、当該機密プログラムとの通信が許可されたユーザー名とプログラム名の組合せを予め規定しておく。

【0019】この場合は、前記機密プログラムと当該機密プログラムがアクセスする機密情報ファイルについても、前記セキュリティ用OSが占有する磁気ディスクに格納すると共に、前記機密プログラムをセキュリティ用OS上のプロセスとして動作させることでサービス用OS側からの不正アクセスから保護する。もしくは、セキュリティ用OSやサービス用OSとは別の第3のOSと、各OS間で通信するためのプロセス間通信手段と、各OSが互いに排他的に占有するメモリや磁気ディスクおよびネットワークデバイスを設け、前記機密プログラムと当該プログラムがアクセスする機密情報ファイルを、前記第3のOSが占有する磁気ディスクに格納すると共に、第3のOS上のプロセスとして動作させることで、サービス用OS側からの不正アクセスから保護する。

【0020】本発明におけるOSとは、ユーザーまたはプログラムからの要求に応じて記憶媒体中のデータ、ファイルへのアクセスを実行する機能と、アクセス要求元のユーザーやプログラムを識別する機能と、排他制御機構により占有する記憶手段を有するプログラムモジュールを意味しており、

- ・データ（ファイル）アクセスを管理しており、検知が可能である。

- ・アクセス要求元のユーザーを識別できる。

- ・アクセス要求元のアプリケーションを識別できる。

【0021】といった特徴を持つ。したがって、一般的にOSと呼ばれるものに限らず上記特徴を持つものであれば、本発明を適用することは可能である。

【0022】また、サービス提供とは、リクエストに応じて所定の処理を実行し、その処理結果を要求元に返すことを指し、また、サービス利用とは、上記サービス提供をおこなうものに対して、リクエストを発行し、その処理結果を受け取ることを指す。

【0023】また、機密情報とは、システム利用者のうち、権限を持つものだけがその内容および存在自体を知り得るような情報を指し、機密プログラムとは、このような情報を扱いながら所定を処理を実行するプログラムを指す。

【0024】本発明を実現するのに必要な各プログラム（コード、モジュール、ユニットともいう）は、ネットワークに接続される他のサーバからコンピュータが読み取り可能な媒体、すなわちネットワーク上の伝送信号、またはCDROM、FDなどの可搬型記憶媒体、を経由して、事前にまたは必要なときに導入してもよい。

【0025】

【発明の実施の形態】以下、図を用いて本発明の実施の

一形態を説明する。

【0026】図1は、本発明のアクセス制御システムの一構成例である。100はサーバ情報処理装置であり、サービス用オペレーティングシステム(OS)103が管理するメモリ101、当該サービス用OS103が占有するディスクコントローラ112aと磁気ディスク114aとLANコントローラ113、セキュリティ用OS104が管理するメモリ102と、当該セキュリティ用OS104が占有するディスクコントローラ112bと磁気ディスク114bその他を備える。

【0027】複数OS制御プログラム116は、サーバ情報処理装置上で複数のOS(本実施例ではサービス用OS103とセキュリティ用OS104)が動作するための各種の制御をする部分で、各ハードウェアの初期化および、メモリと磁気ディスクの分割占有処理、CPUのスケジューリング、割り込み処理等を行う。OS間通信処理部108は、複数OS制御プログラム116の中に存在し、サーバ情報処理装置上で動作するOS間での通信機能を提供する。やサーバ情報処理装置100の起動時には、前記複数OS制御プログラム116が各ハードウェアの初期化およびOS毎のハードウェア分割占有処理等を行い、サービス用OS103とサーバプログラム109がディスクコントローラ112aを介して磁気ディスク114aからメモリ101上にロードされると共に、セキュリティ用OS104とアクセス制御プログラム110がディスクコントローラ112bを介して磁気ディスク114bからメモリ102上にロードされる。前記OS間通信処理部108は、ファイルI/Oフックプログラムと前記アクセス制御プログラム110との間でデータ交換を行なうために使用するものとし、前記サーバプログラム109からは直接使用できないインタフェースとなっている。

【0028】なお、OS間通信処理部108は、サービス用OS103に含まれるプロセス間通信プログラムとして、構成することも可能である。

【0029】また、サービス用OS103は、流通している既存のマルチユーザー・マルチタスクなOSと同様、ユーザーの識別・認証機能を具備しているものとする。前記サーバ情報処理装置100と複数のクライアント情報処理装置120は、ローカルエリアネットワーク(LAN)115を介して相互に接続されており、前記サーバプログラム109は、LANコントローラ113とLAN115を介して、クライアントプログラム121からのリクエストの受信と、当該クライアントプログラム121へのレスポンスの送信を行なう。サーバプログラム109とクライアントプログラム121は、例えばWWWサーバとブラウザに相当するプログラムである。なお、前記サーバ情報処理装置100とクライアント情報処理装置120は、電話回線やインターネットを介して接続されていてもよい。また、サービス用OS1

03とセキュリティOSとは、OS間通信のインタフェースを非公開にすれば別々の装置上にあっても良く、セキュリティ的にも問題はない。

【0030】前記サーバプログラム109のようにOS上で動作するアプリケーションプログラムは、一般にユーザーレベルのプロセスと呼ばれている。該サーバプログラム109は、前述のようにクライアントプログラム121からのリクエストに基づいて処理を実行するものであり、言い換えればネットワークからの攻撃の対象にもなり得るプログラムである。これに対し、ファイルI/Oフックプログラム106やファイルシステムドライバ107のようにOSの一機能として動作するプログラムは、一般にカーネルレベルのモジュールと呼ばれており、多くのコンピュータシステムではアクセス制御機能をカーネルレベルのモジュールとして実装している。

【0031】図1のサーバ情報処理装置100のように、1台の情報処理装置上に複数のOSを同時に動作させるための技術としては、仮想計算機あるいはマイクロカーネルがある。その他、リアルタイムOSを一般的なOSのカーネルレベルのモジュールとして実装すると共に、前記リアルタイムOSがシステム障害を検知すると、リアルタイム処理を継続しながらもシステムを自動的に再起動する方法が、特開平11-024943号公報に開示されている。また、OS間通信処理部108のような、異種OS上のプロセス間通信を実現する方法は、特開平11-085546号公報に開示されている。これらを本発明の前提条件として、引続き実施の形態を述べる。

【0032】図2は、前記ポリシーファイル200のデータ構造を示したものである。

【0033】210から212は、ポリシー記述の一例を示したものである。オブジェクト名201は、アクセスを制限すべきファイルの名称を示す。オブジェクト名として、210や212のようにファイル単位の指定と、211のようなディレクトリ単位の指定が可能となっている。ディレクトリ単位の指定では、該ディレクトリ名に続くファイル名を所定の文字、記号(たとえばアスタリスク(*))で表記する。前記オブジェクトに対して禁止すべきアクセスを、禁止されたアクセスのタイプ202に示す。エラーコード203は、前記禁止されたアクセスのタイプ202が発生した際に、当該アクセス発行元(サブジェクト)となるプログラムに返すべきエラーコードを示す。例外サブジェクト204は、特別にアクセスを許されたプログラムの名称である。プログラムのハッシュ値205は、前記例外サブジェクトとして指定されたプログラムファイルの特徴値(ハッシュ値)をたとえば8バイトで表したものである。ユーザー名206は、前記例外サブジェクト204のプログラムを利用可能なユーザーを、前記サービス用OS103が管理するユーザー名またはグループ名で表したものであ

る。

【0034】つまり前記ポリシーは、前記プログラムのハッシュ値205に登録されたハッシュ値を有するプログラムが前記例外サブジェクトとして登録され、且つ該プログラムをユーザー名206に登録されたユーザーあるいはグループのメンバーが利用している場合に限り、例外としてアクセスを許可することを表している。同時に、これら条件が整っていない場合は、アクセス発行元のプログラムに対して前記エラーコード203を返すことを意味する。

【0035】前記ポリシーファイル200の設定は、システムのセキュリティ管理者が行なうものとする。例えば、HTMLファイルなどのWWWコンテンツを侵入者によって不正に書き換えられないようにするには、該HTMLファイルへのライトアクセスを原則として禁止しておき、例外としてコンテンツ管理者に任命されているユーザーが特定のHTML編集用プログラムを用いた場合に限りライトアクセスを許可するといったポリシーを前記ポリシーファイル200に記述すればよい。

【0036】例外サブジェクトには、サーバプログラム109のように、ネットワークからの攻撃を受けやすいプログラムを指定しないことが重要である。更に例外サブジェクトとして登録するプログラムを、CD-ROM等の取り外し可能な記憶媒体に格納しておき、必要ときだけ媒体を装着して使用すればより確実に保護できるので、さらなるセキュリティ効果が期待できる。図1の構成に当該媒体の駆動装置を追加し、ディスクコントローラ112aが当該駆動装置に対応すれば、このような記憶媒体が使用可能となる。

【0037】図3は、アクセスログファイル300のデータ構造の一例を示したものである。アクセスログファイル300は、前記ポリシーファイル200にて禁じられているアクセスが発生した事実を、アクセス制御プログラム110のアクセスログ登録ルーチン406が書き込むためのファイルであり、当該アクセスが発生した日時301と、当該アクセスの対象となったファイルを表すオブジェクト名302と、当該アクセスのタイプ303、当該アクセスを発行したプログラムを表すサブジェクト名304、そして前記プログラムを利用していたユーザーを表すユーザー名305から構成される。

【0038】図4に、ファイルI/Oフックプログラム106と、アクセス制御プログラム110の構成を示す。

【0039】ファイルI/Oフックプログラム106は、前記サーバ情報処理装置100の起動時に、サービス用OS103と共にメモリ101上にロードされる。アクセス制御プログラム110は、前記サーバ情報処理装置100の起動時に、セキュリティ用OS104と共にメモリ102上にロードされる。

【0040】図5と図6を用いて、本発明のアクセス制

御の概要を記す。

【0041】図5は、I/Oマネージャ105と、ファイルI/Oフックプログラム106と、アクセス制御プログラム110の三者間における処理の流れを示すものである。501は、サーバプログラム109が発行したファイルアクセスが、I/Oマネージャ105を経由してファイルI/Oフックプログラム106に到達するまでの通信経路であり、サービス用OS103の上で動作する全てのアプリケーションプログラムからのファイルアクセスについて共通なものである。ファイルI/Oフックプログラム106には、図6に示すI/Oパケット600のデータが渡される。I/Oパケット600は、I/Oフックプログラム106に渡される前にサービス用OS103が具備するアクセス制御機構をパスしていることと仮定して説明するが、これに限ったものではなく、前記アクセス制御プログラム110のチェックをパスしてからサービス用OS103が具備するアクセス制御機構のチェックを受けるものであってもよい。

【0042】図6は、前記三者間を流れるパケットのデータ構造を示したものである。601はアクセス対象のファイルを表すオブジェクト名である。602は前記オブジェクトへのアクセスタイプを示す。603は、当該アクセスを発行したプログラムのプロセスIDを示す。604は当該アクセスの処理結果を示すステータスを示す。605は当該アクセスに関連するデータの長さを示すものであり、例えば前記アクセスタイプ602がファイルへの書き込み（ライト）要求であれば、書き込みデータがデータ領域607に格納され、当該書き込みデータの長さがデータ長605に格納される。606は、前記データ領域607へのポインタである。

【0043】図5において、502は、ファイルI/Oフックプログラム106が検知したファイルオープン要求に関するアクセス権限チェックと、アクセスログ300への書き込みを、アクセス制御プログラム110に対してリクエストするに用いる通信経路である。前記リクエストのデータ構造を、図6のリクエストパケット610に示す。サブジェクト名611は、ファイルアクセスの発行元となるプログラムの名称であり、ユーザー名612は、前記プログラムの利用者をユーザー名とグループ名で表すものである。サブジェクト名611及びユーザー名612は、受信したI/Oパケット600に含まれるプロセスID603を指定して、前記サービス用OS103として用いる標準のOSに対してシステムコールを発行することで取得できる。

【0044】前記リクエストパケット610は、OS間通信処理部108が当該パケットのデータを、ファイルI/Oフックプログラムのメモリ空間630からアクセス制御プログラムのメモリ空間640に複写することで伝達される。アクセス制御プログラム110では、受信したリクエストパケットに含まれる情報を、ポリシーフ

ファイル200の内容と照合し、その結果をレスポンスパケット620に登録して前記ファイル1/Oフックプログラムに伝達する。この伝達は、前記通信経路502と同様に、OS間通信処理部108が前記パケットのデータを、アクセス制御プログラムのメモリ空間640からファイル1/Oフックプログラムのメモリ空間630に複写することで達成される。503は、当該レスポンスパケットの通信経路を示すものである。

【0045】ファイル1/Oフックプログラム106は、受信したレスポンスパケット620の内容から当該アクセスの可否を判断し、アクセス不可であればエラーコード203を1/Oパケット600のステータス604に設定し、当該1/Oパケットを前記1/Oマネージャ105に返す。前記ステータス604にエラーコードが設定されている場合、前記1/Oマネージャ105は、通信経路506を用いてファイルアクセス発行元のサーバプログラム109に当該エラーを返す。一方、前記ステータス604にエラーコードが設定されていない場合は、通信経路505を用いてファイルシステムドライバ107及びディスクコントローラ112aを介して磁気ディスク114aへのファイルアクセスを続行し、通信経路506を用いて当該アクセスの結果を前記サーバプログラム109に返す。

【0046】本発明のアクセス制御の具体的処理内容を、図7から図16を用いて説明する。

【0047】図7は、ファイル1/Oフックプログラム106が具備するファイル1/Oフックルーチンの処理内容を示すフロー図である。

【0048】サーバプログラム109が、ステップ701でクライアントプログラム121からのリクエストを受信し、ステップ702で前記リクエストに応じたファイルアクセスをサービス用OS103に対して発行した場合を想定する。

【0049】当該ファイルアクセスは、通信経路501を経由して、ファイル1/Oフックルーチン400に1/Oパケット600として渡される（ステップ703）。ステップ703からステップ712は、ファイル1/Oフックルーチン400の処理フローである。ステップ704では、1/Oパケット600に含まれるプロセスID603から、ファイルアクセス発行元のサブジェクト名611とユーザー名612を取得する。

【0050】ステップ705からステップ708では、アクセスタイプ602を調べ、ファイルオープン、ファイルクローズ、ファイルリード又はライト、またはファイル削除又はリネームにそれぞれ対応する処理ルーチンを実行する。これらに該当しない場合、又は対応する処理ルーチンを実行した後は、通信経路504を経由して、ステップ713にて1/Oマネージャ105の処理に戻る。

【0051】図8を用いてオープン処理ルーチン401

の処理フローを説明する。ステップ801では、アクセス日時として現在の日付と時刻を取得する。ステップ802では、リクエストパケット610を作成すると共に、当該パケット中のエラーコード203とハッシュ値205を0に初期設定する。次にステップ803にて、アクセス制御プログラム110のオープン制御ルーチン405をコールすると共に、前記リクエストパケット610をオープン制御ルーチン405に渡す。

【0052】オープン制御ルーチン405の処理フローを、図9を用いて説明する。ステップ901では、受信したリクエストパケット610のうち、オブジェクト名601とアクセスタイプ602の内容を、ポリシーファイル200と照合する。ステップ902では、当該アクセスが禁止されたアクセスのタイプとして登録されているかどうかを判別する。このとき、禁止されたアクセスタイプに該当しなければ正当なアクセスとみなし、ステップ903にて、許可されたアクセスタイプをレスポンスパケット620の認可アクセス613に登録し、オープン処理ルーチン401の処理に戻る。ここで、許可されたアクセスタイプとは、オブジェクト名601で示されるファイルに対するリードとライトの内、サブジェクト名611で示されるプログラムから実行可能なアクセスタイプのことを表す。したがって、オブジェクト名601で示されるファイルに対して、ポリシーファイル200の中でリードとライト共に許可されていれば、認可アクセス613にはリードとライトの両方を登録する。また、リードのみが許可されている場合はリードのみを、ライトのみが許可されている場合はライトのみを前記認可アクセス613に登録する。リードとライトの両方を禁じる場合には、ポリシーとして当該ファイルのオープンを禁止するよう、予めポリシーファイル200に記述しておけばよい。

【0053】ステップ902にて禁止されたアクセスだと判断した場合、ステップ904にて、前記サブジェクト名611とユーザー名612が、共にポリシーファイル200に記述された例外サブジェクト204とユーザー名206に該当するか否かを判別する。このとき、サブジェクト名については該プログラムファイルのパス名が完全に一致するか否かを判別する。またユーザー名については、ユーザー名612に含まれるユーザー名とグループ名のいずれかが一致するか否かを判別する。判別の結果、例外サブジェクト204かつユーザー名206に該当する場合、ステップ905にて当該プログラムファイルのハッシュ値205をポリシーファイル200から取得して、レスポンスパケット620に設定する。また、ステップ906では、ステップ903と同様に、許可されたアクセスタイプをレスポンスパケット620の認可アクセス613として設定する。その後、ステップ908で、該当するエラーコード203をレスポンスパケット620に設定してから、オープン処理ルーチン4

01へ戻る。ここでのエラーコード設定は、オープン処理ルーチン401のステップ804からステップ812の処理の中で意味を持つものである。

【0054】ステップ904にて、例外サブジェクト204とユーザー名206とに該当しないと判断した場合、ステップ907にて、当該アクセスの内容をアクセスログファイル300に書込む。その後、ステップ908で、該当するエラーコード203をレスポンスパケット620に設定してから、オープン処理ルーチン401へ戻る。

【0055】次に、図8において、ステップ804ではレスポンスパケット中のハッシュ値205の値を調べ、0であれば例外として認められたアクセスではないと判断し、ステップ805でエラーコードの値をチェックする。エラーコードが0でなければ禁止されたアクセスであったと判断し、ステップ806にて、I/Oパケット600のステータス604に、当該エラーコード203を設定して処理を終了する。ステップ805で、エラーコードが0であれば、正当なアクセスであったと判断し、ステップ810にて当該アクセス情報を後述するオープンファイルテーブルの先頭アドレスに登録してから処理を終了する。

【0056】ステップ804でハッシュ値が0でない場合は、例外として認められたサブジェクトであると判断し、ステップ807にてサブジェクト名601で示されるプログラムファイルのハッシュ値を算出し、ステップ808にてレスポンスパケット中のハッシュ値205と比較する。両者が等しければ、例外サブジェクト204に相当するとみなして、ステップ812にて当該アクセス情報をオープンファイルテーブルの先頭アドレスに登録してから本ルーチンの処理を終了する。ハッシュ値が等しくなければ、前記プログラムファイルが不正なプログラムであるとみなして、ステップ809にて前記リクエストパケットにおけるサブジェクト名611を例外サブジェクトにならないように、例えばサブジェクト名としてnullデータに、変更し、ステップ810にてアクセスログ登録ルーチン406をコールすると共に前記リクエストパケット610をアクセスログ登録ルーチン406に渡す。

【0057】該アクセスログ登録ルーチン406の処理フローを、図14を用いて説明する。ステップ1401で、リクエストパケット610の内容をポリシーファイル200と照合する。予め前記ステップ809で例外サブジェクト扱いにならぬようにサブジェクト名611をクリアしておくことにより、必ず禁止されたオープンアクセスとして扱われる。ステップ1402にて、該当するエラーコードをポリシーファイル200から読み出してレスポンスパケット620に設定し、当該パケットを返す。ステップ1403にて当該アクセス内容を、アクセスログファイル300に書き込み、オープン処理ルー

チン401の処理に戻る。

【0058】図8のオープン処理ルーチン401において、ステップ811でレスポンスパケット中のエラーコード203をI/Oパケットのステータス604に設定して本ルーチンの処理を終了する。

【0059】図10を用いてオープンファイルテーブルについて説明する。オープンファイルテーブル1000は、現在オープン中のファイルに関する情報を格納した構造体データ1002の集合である。1個の構造体には1件のオープンファイルの情報を記憶しており、ファイルI/Oフックプログラム106では、各構造体を先頭アドレス1001とポインタ1003によりリストとして管理する。本オープンファイルテーブル1000の一例を図11に示す。

【0060】オープンファイルテーブル1000に登録された情報に該当するアクセスであれば、該アクセスを許可することになるため、オープンファイルテーブル1000を不正に書換えられないよう保護することが重要である。サービス用OS103として、プロセス毎に独立したメモリ空間が割当てられると共にメモリ空間の排他制御機構が働くOSを使うことで、前記オープンファイルテーブル1000も別プロセスからは不正に書換えできない仕組みにすることが可能になる。

【0061】図12は、クローズ処理ルーチン402の処理フローを示したものである。ステップ1201では、受信したI/Oパケット600の中にあるオブジェクト名601と、プロセスID603、並びにプロセスID603から取得したサブジェクト名611およびユーザー名612の組み合わせと同じものをオープンファイルテーブル1000から検索する。ステップ1202で、該当する情報がテーブルにあれば、ステップ1203にて該当する情報をテーブルから削除する。一方、テーブルに存在しなければ、ポリシーファイル200に登録されていないファイルへのクローズ要求とみなし、本ルーチンの処理を終了する。

【0062】図13は、リード・ライト処理ルーチン403の処理フローを表したものである。ステップ1301では、受信したI/Oパケット600の中にあるオブジェクト名601と、プロセスID603と、プロセスID603から取得したサブジェクト名611と、ユーザー名612との組み合わせと同じものをオープンファイルテーブル1000から検索し、更にアクセスタイプ602が認可されたアクセスタイプ613に含まれるかをチェックする。ステップ1302にて、前記オブジェクト名601と、プロセスID603と、サブジェクト名611と、ユーザー名612と、アクセスタイプ602との組合せがテーブル1000にあれば、本ルーチンの処理を終了する。このことは、オープンファイルテーブル1000に登録されたアクセスであれば正当なアクセスであるとみなし、アクセス制御プログラム11

0の処理を行わないことを意味する。これは、本発明のアクセス制御によって生じるシステムのパフォーマンス低下を軽減する効果がある。

【0063】ステップ1302にて、オープンファイルテーブルに登録されたアクセスでなければ前記ポリシーファイル200により禁止されたアクセスとみなし、ステップ1303でアクセス日時を取得し、ステップ1304でリクエストパケット610を作成する。このとき、エラーコード203を0に初期設定する。この後、ステップ1305にてアクセスログ登録ルーチン406の処理にジャンプする。

【0064】図14のアクセスログ登録ルーチン406において、ステップ1401ではリクエストパケット610の内容をポリシーファイル200と照合し、ステップ1402にて、該当するエラーコードをポリシーファイル200から読み出してレスポンスパケット620に設定する。次に、ステップ1403にて当該アクセス内容を、アクセスログファイル300に書き込み、リード・ライト処理ルーチン403の処理に戻る。リード・ライト処理ルーチン403では、図13のステップ1306において、レスポンスパケットで受信したエラーコードを1/0パケットのステータス604に設定し、本ルーチンの処理を終了する。

【0065】図15は、削除・リネーム処理ルーチン404の処理フローを表したものである。図8のオープン処理ルーチン401と同じ処理内容のステップには同じ番号を付している。

【0066】ステップ1503にて、アクセス制御プログラム110の削除・リネーム制御ルーチン407をコールすると共に、前記リクエストパケット610を削除・リネーム制御ルーチンに渡す。

【0067】削除・リネーム制御ルーチン407の処理フローを、図16を用いて説明する。図9のオープン制御ルーチン405と同じ処理内容のステップには同じ番号を付している。ステップ1602では、当該アクセスが禁止されたアクセスのタイプとして登録されているかどうかを判別する。禁止されたアクセスタイプに該当しなければ正当なアクセスとみなし、削除・リネーム処理ルーチン404の処理に戻る。

【0068】ステップ1602にて禁止されたアクセスだと判断した場合、図9と同様にステップ904、905の処理を行い、ステップ1603にて、例外サブジェクト204とユーザー名206に該当しないと判断した場合、図9と同様にステップ907の処理を行う。

【0069】ステップ905または907の処理を済ませると、ステップ908で、該当するエラーコード203をレスポンスパケット620に設定してから、削除・リネーム処理ルーチン404へ戻る。

【0070】次に図15において、図8と同様にステップ804の処理を行う。ステップ1505では、エラー

コードが0であれば、正当なアクセスであったと判断し、本ルーチンの処理を終了する。ステップ807にてサブジェクト名601で示されるプログラムファイルのハッシュ値を算出し、ステップ808にてレスポンスパケット中のハッシュ値205と比較して、両者が等しければ、例外サブジェクト204に相当するとみなして本ルーチンの処理を終了する。ハッシュ値が等しくない場合は図8と同様の処理を行い、アクセスログ登録ルーチン406を呼び出す。当該ルーチンが処理を済ませると削除・リネーム処理ルーチン404の処理に戻る。

【0071】図15において、ステップ1511でレスポンスパケット中のエラーコード203を1/0パケットのステータス604に設定して本ルーチンの処理を終了する。

【0072】以上、本発明の実施の一形態を説明したが、OSが備えるファイルシステムの種類によっては、ファイルオープン要求を表す1/0パケットの中に、目的となるアクセスタイプを付属情報として含むものがある。つまり、ファイルのリードアクセスをするためのオープン要求なのか、リード・ライトのためのオープン要求なのかを、前記1/0パケット600中のアクセスタイプ602から検知できる。このようなファイルシステムを前提とした場合には、図7に示すファイル1/0フックルーチン400の処理フローと、図8に示すオープン処理ルーチン401の処理フローと、更に図9に示すオープン制御ルーチン405の処理フローとを変更することにより、前記オープンファイルテーブル1000の管理が不要なアクセス制御システムが実現できる。

【0073】図7においてはファイル1/0フックルーチン400の処理フローから、ステップ706と、ステップ707と、ステップ710と、ステップ711を省く。つまり、ファイルオープン要求と、ファイル削除要求と、ファイルのリネーム要求のいずれかを検知し、ファイルのオープン要求であれば、ステップ709にてオープン処理ルーチン401の処理に移り、削除又はリネーム要求であれば、ステップ712にて削除・リネーム処理ルーチン404の処理に移るよう、ファイル1/0フックルーチン400の処理フローを変更すればよい。

【0074】図8においては、オープン処理ルーチンの処理フローから、ステップ812の処理を省く。つまり、ステップ808でのハッシュ値照合処理の結果、等しければ本ルーチンの処理を終了するよう、オープン処理ルーチン401の処理フローを変更する。

【0075】更に、図9で説明したオープン制御ルーチン405の処理フローは、図26に示す処理フローに変更する。図26のステップ2801における処理、すなわち、受信したリクエストパケット610のうち、オブジェクト名601とアクセスタイプ602の内容を、ポリシーファイル200と照合するといった処理において、このときアクセスタイプ602の中に含まれる前記

オープン要求の付属情報（リードやライト）まで取得して、前記ポリシーファイル200と照合する。本発明では、前記オープン要求の付属情報として、リードとライトの両方が含まれる場合、リードとライトの両方、もしくはいずれか一方が禁止されたアクセスとしてポリシーファイル200に記述されていれば、当該オープン要求を禁止されたアクセスとしてみなす。

【0076】ステップ2802では、当該アクセスが禁止されたアクセスのタイプとして登録されているかどうかを判別する。このとき、禁止されたアクセスタイプに該当しなければ正当なアクセスとみなし、ステップ2803にて、レスポンスパケット620のエラーコード203と、ハッシュ値205を共に0を設定し、オープン処理ルーチン401の処理に戻る。リードとライトの両方を禁じる場合には、ポリシーとして当該ファイルのオープンを禁止するよう、予めポリシーファイル200に記述しておけばよい。

【0077】ステップ2802にて禁止されたアクセスだと判断した場合、ステップ2804にて、前記サブジェクト名611とユーザー名612が、共にポリシーファイル200に記述された例外サブジェクト204とユーザー名206に該当するか否かを判別する。このとき、サブジェクト名については該プログラムファイルのパス名が完全に一致するか否かを判別する。またユーザー名については、ユーザー名612に含まれるユーザー名とグループ名のいずれかが一致するか否かを判別する。判別の結果、例外サブジェクト204かつユーザー名206に該当する場合、ステップ2805にて当該プログラムファイルのハッシュ値205をポリシーファイル200から取得して、レスポンスパケット620に設定する。その後、ステップ2808で、該当するエラーコード203をレスポンスパケット620に設定してから、オープン処理ルーチン401に戻る。

【0078】ステップ2804にて、例外サブジェクト204とユーザー名206とに該当しないと判断した場合、ステップ2807にて、当該アクセスの内容をアクセスログファイル300に書込む。その後、ステップ2808で、該当するエラーコード203をレスポンスパケット620に設定してから、オープン処理ルーチン401に戻る。

【0079】次に、本発明のアクセス制御システムのうち、プロセス間通信に関するアクセス制御について説明する。図17は、機密性の高い情報と、それにアクセスするプログラムを不正アクセスから保護するためのアクセス制御システムの一構成例であり、図1と同じ構成要素には、同じ番号を付している。この構成は、情報の機密性を確保しつつ、外部ネットワークに対してサービスを提供する場合に有効であり、高い安全性を確保することができる。

【0080】図17の構成は、図1の構成と同様である

が、さらに、機密用OS122が管理するメモリ1708と、当該機密用OS122が占有するディスクコントローラ112cと磁気ディスク114cとLANコントローラ123その他を備える。

【0081】前記サービス用OS103が占有するLANコントローラ113aは、インターネット等の外部ネットワーク1706に接続されたクライアント情報処理装置120等との通信に用いる。一方、前記機密用OS122が占有するLANコントローラ113cは、内部ネットワーク1707に接続された機器との通信に用いる。

【0082】複数OS制御プログラム116は、機密用OS122まで含めた複数のOSの制御を行う。

【0083】前記サービス用OS103とセキュリティ用OS104と機密用OS122は、各々ドライバ部117とドライバ部118とドライバ部119を含んでおり、各OS上のアプリケーションプログラムは、必ずこれらドライバを経由してOS間通信処理部108へアクセスする。この方法は、すでに説明したプロセス間通信方法と同様なものである。

【0084】一般的なプロセス間通信の種類には、共有メモリやセマフォ、メッセージキューなど様々なものがあり、いずれも予め取り決められた名称や番号を用いて通信相手を識別することになっている。

【0085】更に本発明は、以下の特徴を持つ。すなわち、プロセス間通信の不正利用を防止するために、プロセス間通信に関するセキュリティポリシーを記述した通信制御ポリシー1800を前記磁気ディスク114bに格納しておき、前記OS間通信処理部108を利用したプロセス間通信の要求については、セキュリティ用OS104側で動作する通信制御プログラム1700が、前記通信制御ポリシー1800の記述に基づいてアクセス制御を行う。また、当該プロセス間通信に関するアクセスログを、前記磁気ディスク114b中の通信ログ1900に記録する機能を備える。また、前記通信制御プログラム1700は、実行中のプロセス間通信に関する情報を、セキュリティ用OS104が管理するメモリ102の中に設けた通信管理テーブル2000にて保持する。

【0086】中継プログラム1701は、前記サービス用OS103の上で動作するアプリケーションプログラムの1つである。本実施例では、外部ネットワークに接続されたクライアント情報処理装置120からの要求に応じて、後述する機密プログラム1703とのプロセス間通信を行ない、処理結果を前記クライアント情報処理装置120に返信する役割をもったプログラムを中継プログラムと称す。なお、前記クライアント情報処理装置120とのデータ通信をサーバプログラム109により実施し、前記中継プログラム1701は、必要に応じてサーバプログラム109が呼び出して利用するものであ

ってもよい。

【0087】機密プログラム1703は、機密用OS122の上で動作するアプリケーションプログラムの1つであり、本実施例では、機密用OS122が占有するディスクコントローラ112cを利用して磁気ディスク114cに格納された機密情報1704にアクセスしたり、同じく機密用OS122が占有するLANコントローラ113cを経由して、占有する内部ネットワークに接続された記憶媒体中の機密情報1705にアクセスしながらサービスを提供するものと仮定して説明する。

【0088】前記サーバ情報処理装置100の起動時には、前記複数OS制御プログラム116が各ハードウェアの初期化、メモリ領域の分割、CPUのスケジューリング等を行う。

【0089】次に、サービス用OS103とサーバプログラム109と中継プログラム1701がディスクコントローラ112aを介して磁気ディスク114aからメモリ101上にロードされ、セキュリティ用OS104とアクセス制御プログラム110がディスクコントローラ112bを介して磁気ディスク114bからメモリ102上にロードされ、更に機密用OS122と機密プログラム1703がディスクコントローラ112cを介して磁気ディスク114cからメモリ1708上にロードされる。

【0090】図18は、前記通信制御ポリシー1800のデータ構造の一例を示したものである。アプリケーション名1801は、機密用OS122の上で動作する機密プログラムの名称であり、機密用OS122上でユニークな識別子となるよう、例えばプログラムファイルの絶対パス名等を用いる。サービス番号1802は、前記アプリケーション名1801に対応する機密プログラムが提供するサービスの識別子であり、例えばTCP/IP通信におけるポート番号や、メッセージ通信におけるメッセージキュー番号（メッセージキュー名称）等に相当する。本実施の形態では、1つの機密プログラムに対して1つのサービス番号を登録する場合を例に説明するが、これに限ったのではなく、複数のサービス番号を登録してもよい。サービス番号は、通信する双方で事前に取り決めておくものとし、中継プログラム1701から通信制御ポリシー1800の参照はできないものとする。

【0091】許可アプリケーション名1803は、サービス用OS103上で動作するプログラムであり、且つ前記アプリケーション名1801で示される機密プログラムとの通信を許可されたプログラムの名称であり、例えば中継プログラム1701に相当する。特徴値1804は、前記許可アプリケーション名1803で示されるプログラムが備える特徴値であり、例えば当該プログラムファイルのサイズやハッシュ値等に相当する。ユーザー名1805は、前記許可アプリケーション名1803

で示されるプログラムが実行中に使用する権限をユーザー名で表したものである。これは、一般的なマルチユーザー・マルチタスクのOSにおいて、アプリケーションプログラムが必ず何れかのユーザー権限を使用して動作することを前提としている。特にユーザー名を限定したくない場合は、ユーザー名を所定の文字、記号（たとえばアスタリスク（*））で表記する。なお、本通信制御ポリシー1800は、システムの運用に先立って、管理者が予め登録しておくものである。また、図18では機密プログラムと中継プログラムの組み合わせが1対1の場合を例に記述しているが、1つの機密プログラムの利用許可を、複数の中継プログラムに与えても良い。

【0092】図19は、前記通信ログ1900のデータ構造の一例を示したものである。通信ログ1900は、前記通信制御ポリシー1800に違反する通信が発生した事実を書き込むためのファイルであり、当該通信が発生した日時1901と、当該通信の対象となった機密プログラム名1902と、機密プログラムとの通信要求を発行したサービス用OS103側プログラムを表すサブジェクト名1903、そしてエラーの内容を表すエラー番号1904から構成される。

【0093】図20は、通信管理テーブル2000のデータ構造の一例を示したものである。通信管理テーブル2000は、機密プログラムの識別子2001と、当該機密プログラムが提供するサービスの番号2002と、機密プログラムとのプロセス間通信のために使用する暗証データ2003と、サービス用OS103のドライバ部117が提供する通信中のプログラムの識別子2004、とから構成される。図20では、各識別子にそれぞれプログラム名称を用いているが、例えば機密用OS122とサービス用OS103が各々管理するプロセスIDのような、プロセス特有の情報と組み合わせて管理してもよい。

【0094】図21から図25は、中継プログラム1701と機密プログラム1703が、前記OS間通信処理部108の通信機能を利用してプロセス間通信をする際の、処理フローを示したものである。本実施の形態において、中継プログラム1701と機密プログラム1703は、クライアントとサーバの関係にあることから、前もって機密プログラム1703がデータ受信可能な状態になるものとして説明する。また、プロセス間通信手段には様々な種類があるが、どの手段を用いた場合でもその手順は、ハンドルの取得処理、データ送信処理とデータ受信処理、ハンドルの解放処理に大別できる。ハンドルとは、アプリケーションプログラムがファイル、メモリなどのオブジェクトをアクセスする際にアクセス対象となるオブジェクトを指定するデータであり、OSから与えられる。ハンドルは、オブジェクトごとにシステム内でユニークな数値であることが保証されている。この場合のオブジェクトとは、例えばファイルやソケットな

ど、データ処理の際に用いるアクセス対象全般を指す。本発明では、機密用OS122やサービス用OS103上のプログラムがハンドル値を取得する際に、各OSのドライバ部を経由して必ず通信制御プログラム1700によるポリシーチェックを受け、正当性を確認するところに特徴がある。

【0095】図21は、前記OS間通信処理部108を介して、機密プログラム1703が通信制御プログラム1700に対してサービスを登録し、中継プログラム1701からのデータ受信待ち状態になるまでの処理フローを示したものである。ステップ2100にて、前記通信制御プログラム1700は、OS間通信処理部108の機能を用いて機密用OS122側からのデータ受信が可能な状態となっている。機密プログラム1703は、ステップ2101にて暗証データを作成し、ステップ2102にて、機密用OS122のドライバ部119に対してサービス番号と暗証データを送付し、中継プログラム1701とのプロセス間通信に利用するハンドルを要求する。

【0096】機密用OS122のドライバ部119は、ステップ2103にて通信制御プログラム1700との通信路を確立し、ステップ2104にて前記機密プログラム1703の識別子とサービス番号と暗証データを、通信制御プログラム1700に送付する。

【0097】通信制御プログラム1700は、ステップ2105にて通信制御ポリシー1800を参照し、機密プログラムとサービス番号が登録されているかどうかを調べ、正当性を確認する。通信制御ポリシー1800に記述されていれば、ステップ2106にて前記機密プログラム1703の識別子情報とサービス番号と暗証データを、通信管理テーブル2000に登録する。通信制御ポリシー1800に記述されていなければ、ステップ2107にてその旨を前記ドライバ部119に返信するためのエラー設定を行う。ステップ2108では、通信制御ポリシーとの照合結果を返信するとともに、ドライバ部118を介して、ドライバ部119との通信路を解放する。

【0098】前記ドライバ部119は、ステップ2109にて通信制御プログラム1700からの返信データをチェックし、通信管理テーブル2000への登録に成功してれば、ステップ2111にて中継プログラム1701と通信するために利用するハンドルをOS間通信処理部108から取得し、ステップ2112にて前記機密プログラム1703への返信データとして、前記ハンドル値を設定する。一方、通信管理テーブルへの登録に失敗した場合は、ステップ2110にて、前記機密プログラム1703への返信データとしてエラーを設定する。

【0099】機密プログラム1703は、ステップ2113にてドライバ部119からの返信データをチェックする。要求したハンドル取得に成功してれば、ステッ

プ2114にてサービス用OS103側からのデータ受信待ち状態になり、ハンドル取得に失敗してれば、ステップ2115にて所定のエラー処理を実行する。

【0100】図22は、前記機密プログラム1703がプロセス間通信に利用していたハンドルを解放する際の処理フローを示したものである。ステップ2200にて、前記通信制御プログラム1700は、OS間通信処理部108の機能を用いて機密用OS122側からのデータ受信が可能な状態となっている。機密プログラム1703は、ステップ2201にて、機密用OS122のドライバ部119に対してハンドル値を指定し、当該ハンドルの解放を要求する。

【0101】機密用OS122のドライバ部119は、ステップ2202にて通信制御プログラム1700との通信路を確立し、ステップ2203にて前記機密プログラム1703の識別子とサービス番号を、通信制御プログラム1700に送付する。

【0102】通信制御プログラム1700では、ステップ2204にて通信管理テーブル2000を参照し、登録された機密プログラムであるかを確認する。通信管理テーブル2000に記述されていれば、ステップ2205にて前記機密プログラムの情報を、通信管理テーブル2000から削除する。通信管理テーブル2000に記述されていなければ、ステップ2206にてその旨を前記ドライバ部119に返信するためのエラー設定を行う。ステップ2207では、通信管理テーブル2000との照合結果を返信するとともに、ドライバ部119との通信路を解放する。

【0103】前記ドライバ部119は、ステップ2208にて通信制御プログラム1700からの返信データをチェックし、通信管理テーブルからの削除に成功してれば、ステップ2210にて今度は中継プログラム1701との通信に利用していたハンドルの解放処理を実行する。一方、通信管理テーブルからの削除に失敗した場合は、ステップ2209にて、前記機密プログラム1703への返信データとしてエラーを設定する。

【0104】機密プログラム1703は、ステップ2211にてドライバ部119からの返信データをチェックし、ハンドル解放に成功してれば、ステップ2212にて処理を終了する。一方、ハンドル解放に失敗してれば、ステップ2213にて所定のエラー処理を実行する。

【0105】図23と図24は、前記OS間通信処理部108を介して、中継プログラム1701が機密プログラム1703とのプロセス間通信用のハンドルを取得するまでの処理フローを示したものである。図23のステップ2300にて、前記通信制御プログラム1700は、OS間通信処理部108の機能を用いてサービス用OS103側からのデータ受信が可能な状態となっている。中継プログラム1701は、ステップ2301に

て、サービス用OS103のドライバ部117に対して利用したいサービス番号を送付し、機密プログラムとのプロセス間通信に利用するハンドルを要求する。サービス番号は、上述の通り、ポート番号と同様に、事前に取り決めておくものとする。参照のために、プログラムに固定的に埋め込んでおくか、定義ファイルに記述しておくなどが考えられる。

【0106】サービス用OS103のドライバ部117は、ステップ2302にて通信制御プログラム1700との通信路を確立し、ステップ2303にて前記中継プログラム1701の識別子とその特徴値、および中継プログラム1701が使用しているユーザー権限を表すユーザー名を取得し、利用したいサービス番号と共に通信制御プログラム1700に送付する。

【0107】通信制御プログラム1700では、ステップ2304にて通信管理テーブル2000と通信制御ポリシー1800を参照し、指定されたサービスが有効であり、且つ当該サービスの利用を許可されたプログラムか否かを調べ、正当性を確認する。具体的には、ドライバ部117から渡された中継プログラムの識別子と特徴値とユーザー名とが、通信制御ポリシー1800の内容に一致すればポリシーに合致していると言える。もしサービスが有効で、且つポリシーに合致していれば、ステップ2305にて前記中継プログラム1701の識別子情報を、前記サービス利用中のプログラムとして通信管理テーブル2000に登録し、該当するサービスの利用に必要な暗証データを中継プログラムに返信するために、通信管理テーブル2000から取得する。一方、サービスが有効でない、あるいはポリシーに合致していなければ、ステップ2306にてその旨を通信ログ1900に書き込むと共に、前記ドライバ部117に返信するためのエラー設定を行う。ステップ2307では、通信制御ポリシーとの照合結果を返信するとともに、ドライバ部117との通信路を解放する。

【0108】前記ドライバ部117は、ステップ2308にて通信制御プログラム1700からの返信データをチェックし、通信管理テーブル2000への登録に成功していれば、つまり中継プログラム1701の認証に成功していれば、ステップ2310にて今度は機密プログラムとの通信用ハンドルを取得する処理(図24)に移る。一方、通信管理テーブルへの登録に失敗した場合は、ステップ2309にて、前記中継プログラム1701にエラーを返して終了する。

【0109】図24では、図21のステップ2114で示したように、前記機密プログラム1703がサービス用OS103側からデータ受信可能であることを前提にして、中継プログラム1701の通信用ハンドル取得処理を説明する。サービス用OS103のドライバ部117は、OS間通信処理部108の機能を利用して、機密プログラム1703との通信用ハンドルを取得する。ス

テップ2402では、先に図23のステップ2307にて通信制御プログラム1700から受信した暗証データを、機密用OS122のドライバ部119を経由して機密プログラム1703に送付する。

【0110】機密プログラム1703では、ステップ2403にて、ドライバ部117から受信した暗証データと、機密プログラム1703が作成した(図21のステップ2101)暗証データとを照合し、結果を前記ドライバ部117に返信する。

【0111】前記ドライバ部117は、ステップ2404にて機密プログラム1703からの返信データをチェックする。前記暗証データが一致している場合には、前記機密プログラム1703との通信を許可されたと判断し、ステップ2405にて、中継プログラム1701への返信データとして、機密プログラム1703との通信ハンドル値を設定する。一方、前記暗証データに誤りがあった場合は、ステップ2406にて、機密プログラム1703との通信用ハンドルを解放すると共に、前記中継プログラム1701への返信データとしてエラーを設定する。

【0112】中継プログラム1701は、ステップ2407にてドライバ部117からの返信データをチェックし、ハンドル取得に成功していれば、以後、OS間通信処理部108の機能を利用して機密プログラム1703とのデータ通信が可能となる(ステップ2408)。一方、ハンドル取得に失敗していれば、ステップ2409にて所定のエラー処理を実行する。

【0113】図25は、前記中継プログラム1701が機密プログラムとのプロセス間通信に利用していたハンドルを解放する際の処理フローを示したものである。ステップ2500にて、前記通信制御プログラム1700は、OS間通信処理部108の機能を用いてサービス用OS103側からのデータ受信が可能な状態となっている。中継プログラム1701は、ステップ2501にて、サービス用OS103のドライバ部117に対してハンドル値を指定して、当該ハンドルの解放を要求する。

【0114】サービス用OS103のドライバ部117は、ステップ2502にて通信制御プログラム1700との通信路を確立し、ステップ2503にて前記中継プログラム1701の識別子とサービス番号を、通信制御プログラム1700に送付する。

【0115】通信制御プログラム1700では、ステップ2504にて通信管理テーブル2000を参照し、当該テーブルに登録された中継プログラムであるかを確認する。通信管理テーブル2000に登録されていれば、ステップ2505にて前記中継プログラムに該当する情報を、通信管理テーブル2000から削除する。通信管理テーブル2000に記述されていなければ、ステップ2506にてその旨を前記ドライバ部117に返信する

ためのエラー設定を行う。ステップ2507では、通信管理テーブル2000との照合結果を返信するとともに、ドライバ部117との通信路を解放する。

【0116】前記ドライバ部117は、ステップ2508にて通信制御プログラム1700からの返信データをチェックし、通信管理テーブルからの削除に成功していれば、ステップ2510にて今度は機密プログラム1703との通信に利用していたハンドルの解放処理を実行する。一方、通信管理テーブルからの削除に失敗した場合は、ステップ2509にて、前記中継プログラム1701への返信データとしてエラーを設定する。

【0117】中継プログラム1701は、ステップ2511にてドライバ部117からの返信データをチェックし、ハンドル解放に成功していれば、ステップ2512にて処理を終了する。一方、ハンドル解放に失敗していれば、ステップ2513にて所定のエラー処理を実行する。

【0118】以上説明したように、本発明によれば、侵入者が強い権限をもつユーザーに成りすました場合でも、ファイルへのアクセスを制限できるという効果がある。

【0119】また、不正なファイルアクセスを未然に防ぐことができるという効果がある。

【0120】また、機密性の高い情報へのアクセス手段となるプログラム自体を保護できるという効果がある。また、前記ポリシーファイルとアクセス制御手段を、外部からの不正アクセスや攻撃から保護できるという効果がある。

【0121】また、図1に示すシステムの、サーバ情報処理装置とクライアント情報処理装置とを結ぶLAN115や、図17に示すシステムの、サーバ情報処理装置とクライアント情報処理装置とを結ぶ外部ネットワーク1706などの通信回線上にファイアウォールを設け、さらに本発明を併用すれば、より強固なセキュリティを確保することが出来る。

【0122】

【発明の効果】本発明によれば、情報処理装置が管理するファイル、情報、および実行中のプロセスを不正なアクセスから保護することが可能になる。

【図面の簡単な説明】

【図1】本発明の実施の形態におけるアクセス制御システムの一構成例を示す図。

【図2】アクセス制御ポリシーの設定を格納するためのポリシーファイルを示す図。

【図3】不正アクセスが発生した事実を記録するためのアクセスログファイルを示す図。

【図4】ファイルI/Oフックプログラム106とアクセス制御プログラム110を構成するプログラムルーチンを示す図。

【図5】本発明の実施の一形態において、各モジュール

間を流れるデータの通信経路を示す図。

【図6】本発明の実施の一形態において、各モジュール間を流れるデータの構造を示す図。

【図7】ファイルI/Oフックルーチン400の処理のフローチャートを示す図。

【図8】オープン処理ルーチン401の処理のフローチャートを示す図。

【図9】オープン制御ルーチン405の処理のフローチャートを示す図。

【図10】オープンファイルテーブルの構造体とそのリストを示す図。

【図11】オープンファイルテーブルに登録されるデータの一例を示す図。

【図12】クローズ処理ルーチン402の処理のフローチャートを示す図。

【図13】リード・ライト処理ルーチン403の処理のフローチャートを示す図。

【図14】アクセスログ登録ルーチン406の処理のフローチャートを示す図。

【図15】削除・リネーム処理ルーチン404の処理のフローチャートを示す図。

【図16】削除・リネーム処理ルーチン407の処理のフローチャートを示す図。

【図17】プロセス間通信に関するアクセス制御システムの一構成例を示す図。

【図18】プロセス間通信に関するポリシーの設定を格納するためのファイルを示す図。

【図19】不正なプロセス間通信が発生した事実を記録するためのログファイルを示す図。

【図20】通信管理テーブルのリストを示す図。

【図21】機密用OS122上のプログラムによる通信用ハンドル取得処理のフローチャートを示す図。

【図22】機密用OS122上のプログラムによる通信用ハンドル解放処理のフローチャートを示す図。

【図23】サービス用OS103上のプログラムによる通信用ハンドル取得処理のフローチャート（前半）を示す図。

【図24】サービス用OS103上のプログラムによる通信用ハンドル取得処理のフローチャート（後半）を示す図。

【図25】サービス用OS103上のプログラムによる通信用ハンドル解放処理のフローチャートを示す図。

【図26】オープン制御ルーチン405の処理のフローチャート（その2）を示す図。

【符号の説明】

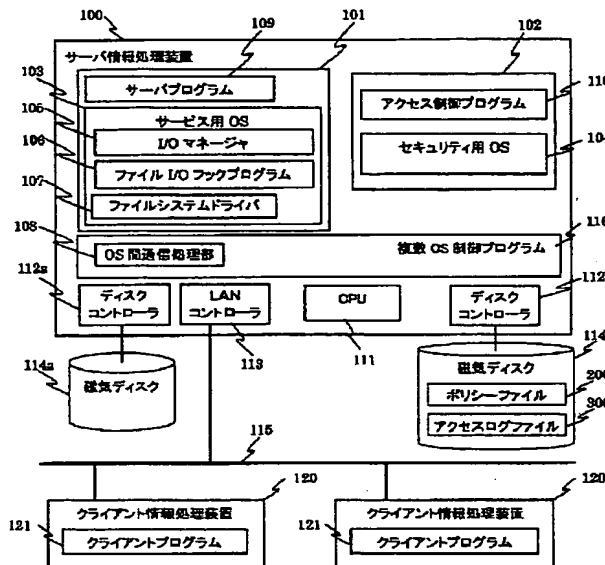
100…サーバ情報処理装置、101…サービス用OSが管理するメモリ、102…セキュリティ用OSが管理するメモリ、103…サービス用OS、104…セキュリティ用OS、105…I/Oマネージャ、106…ファイルI/Oフックプログラム、107…フ

ファイルシステムドライバ、108…OS間通信制御部、109…サーバプログラム、110…アクセス制御プログラム、111…CPU、112…ディスクコントローラ、113…LANコントローラ、114…磁気デバイス、115…LAN、116…複数OS制御プログラム、117～119…ドライバ部、120…クライアント情報処理装置、121…クライアントプログラム、200…ポリシーファイル、300…アクセスログファイル、501～506…通信経路、6

00…I/Oパケット、610…リクエストパケット、620…レスポンスパケット、1000…オープンファイルテーブル、1700…通信制御プログラム、1701…中継プログラム、1703…機密プログラム、1704～1705…機密情報、1706…外部ネットワークとの接続回線、1707…内部ネットワークとの接続回線、1708…機密用OSが管理するメモリ、1800…通信制御ポリシー、1900…通信ログ、2000…通信管理テーブル

【図1】

図1

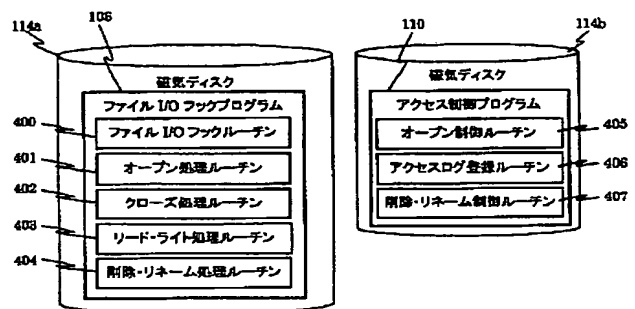


100: サーバ情報処理装置
101: サービス用 OS が管理するメモリ
102: セキュリティ用 OS が管理するメモリ

120: クライアント情報処理装置

【図4】

図4



114a, 114b: 磁気ディスク

【図3】

図3

日時	オブジェクト名	アクセスタイプ	サブジェクト名	ユーザー名
1999.07.28 15:32:46	D:\DOC\SECRET.TXT	Write	c:\prog\wwwsrv.exe	u_0023
1999.07.27 09:15:10	D:\DOC\SECRET.TXT	Delete	c:\prog\wwwsrv.exe	u_0023
1999.07.25 16:02:55	D:\SYS\CONFIG\	Write	c:\prog\txtdit.exe	intruder
1999.07.25 14:44:28	D:\SYS\CONFIG\	Delete	c:\prog\fileman.exe	intruder
1999.07.25 14:42:59	D:\LOG\LOG.TXT	Read	c:\prog\txtdit.exe	user007
1999.07.16 10:29:31	D:\LOG\LOG.TXT	Delete	c:\prog\fileman.exe	user007

300: アクセスログファイル

【図2】

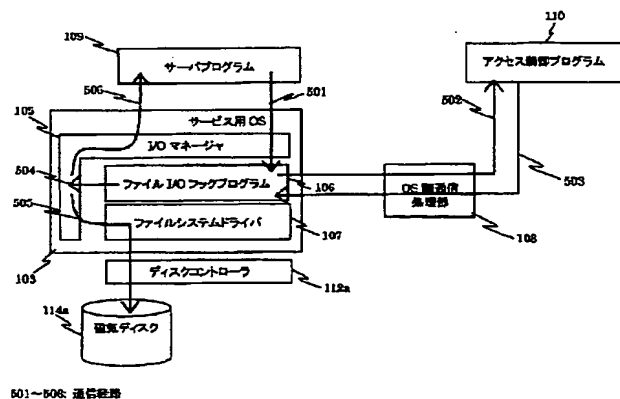
図2

	200	201	202	203	204	205	206
	オブジェクト名	禁止されたアクセスのタイプ	エラーコード	例外サブジェクト	プログラムのハッシュ値	ユーザー名	
210	D:\DOC\SECRET.TXT	Open	0016	c:\prog\wordedit.exe	0x22F0A412B73209CC	sec_admin	
211		Delete	0021	c:\prog\fileman.exe	0x73209C2F0A212B4C	sec_admin	
	D:\SYS\CONFIG*	Write	0018	c:\prog\mantool.exe	0xB74122209FDE236C	eye_admin	
		Delete	0021	c:\prog\mantool.exe	0xB74122209FDE236C	eye_admin	
212	D:\LOG\LOG.TXT	Read	0015	c:\prog\audit.exe. c:\prog\vcck.exe	0x2F21204B73C09 A2C 0xF0A271243B9C202C	root, system	
		Rename	0024	c:\prog\audit.exe	0x2F21204B73C09 A2C	root	

200: ポリシーファイル

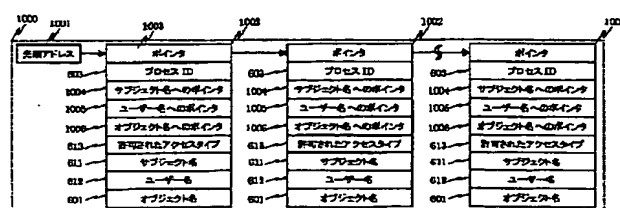
【図5】

図5



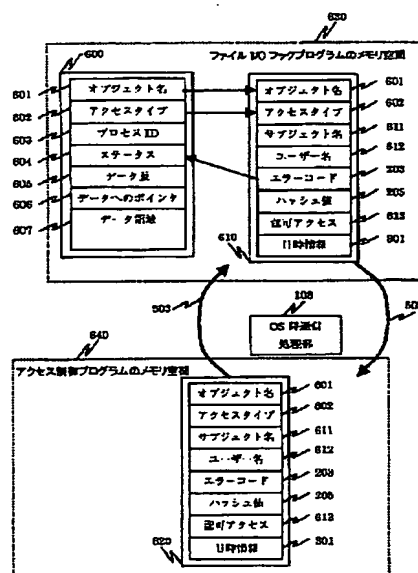
【図10】

図10



【図6】

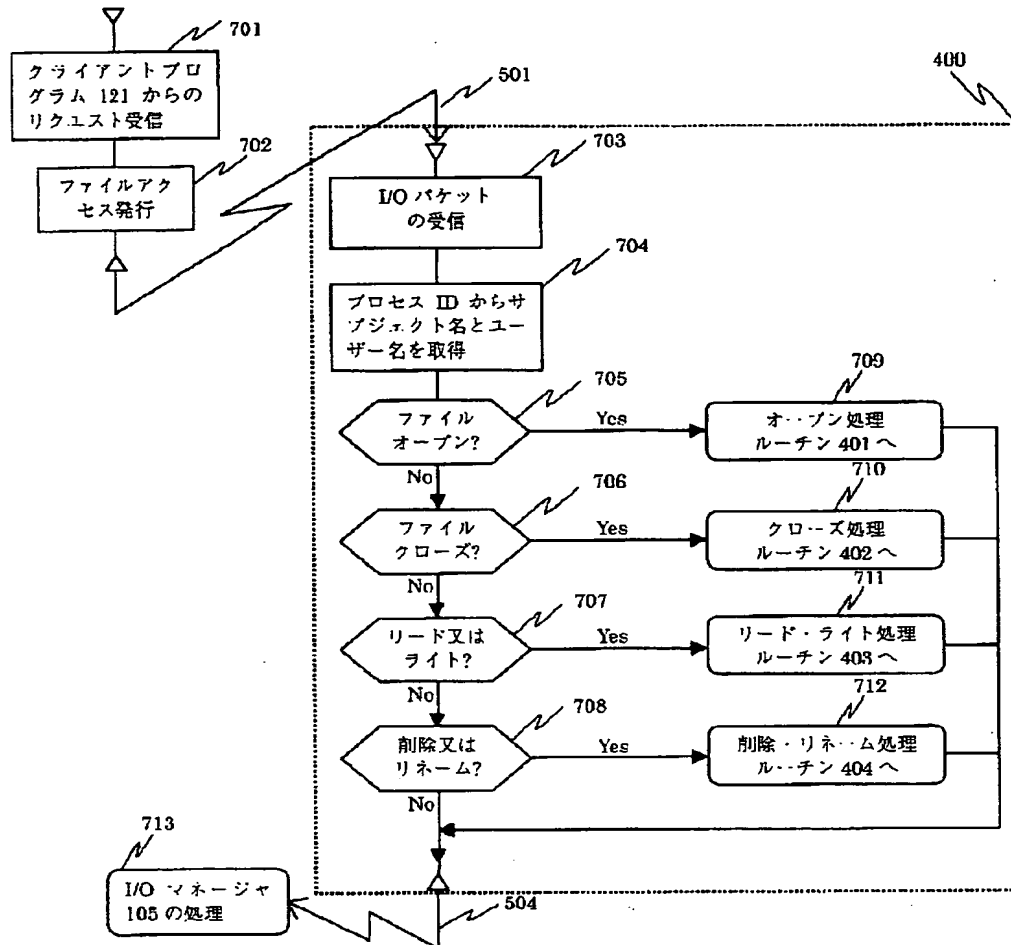
図6



600: IO パケット
610: リクエストパケット
620: レスポンスパケット

【図7】

図7



701~702: サーバプログラム 109 の概略処理フロー

703~712: ファイル I/O フックルーチン 400 の処理フロー

【図11】

図11

プロセスID	サブジェクト名	ユーザー名	オブジェクト名	認可された アクセスタイプ
0022	c:\prog\wwwserv.exe	inet	D:\HOME\SALES.HTML	Read
0043	c:\prog\wordedit.exe	sec_admin	D:\DOC\SECRET.TXT	Read/Write
0067	c:\prog\viewer.exe	taxou	D:\DOC\VIDEA.TXT	Read
0092	c:\prog\manuool.exe	sys_admin	D:\SYS\CONFIG\PORT	Read/Write
0113	c:\prog\rvck.exe	system	D:\DOC\SECRET.DOC	Read
0218	c:\prog\randit.exe	root	D:\LOG\LOG.TXT	Read/Write

【図19】

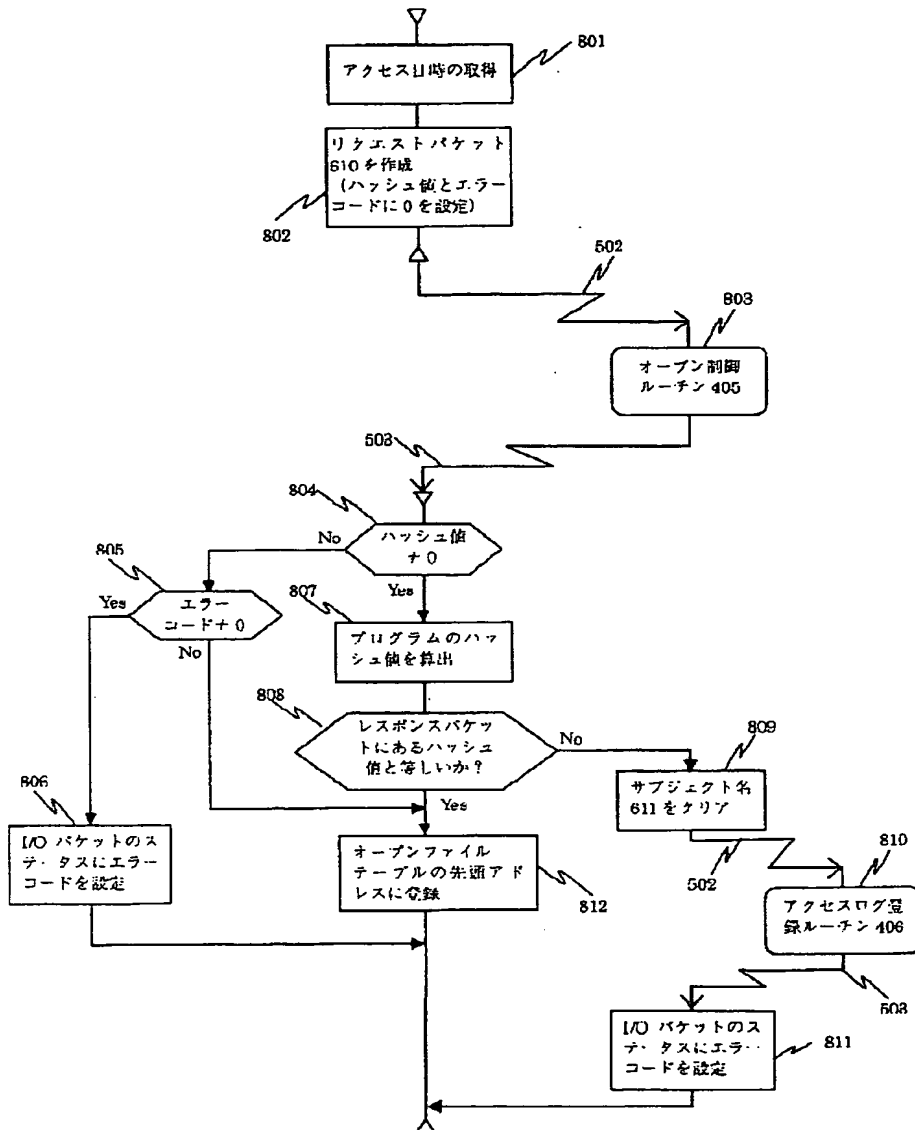
図19

日時	検索プログラム名	サブジェクト名	エラー番号
2000.06.29 18:13:49	/bin/prog/login	c:\prog\cgi\login.exe	0x00CC
2000.06.29 03:44:06	/bin/prog/sandaka	c:\prog\cgi\zand.exe	0x0099
2000.07.05 03:13:12	/bin/prog/shinsei	c:\prog\cgi\shin.exe	0x00A0

1900: 通信ログ

【図8】

図8



801～810: オープン処理ルーチンの処理フロー

【図18】

図18

1800	1801	1802	1803	1804	1805
アプリケーション名	サービス番号	許可アプリケーション名	特徴値	ユーザー名	
/bin/prog/login	30	c:\prog\cgi\login.exe	0xA432B7CC	www	
/bin/prog/zandaka	80	c:\prog\cgi\zand.exe	0x2876FA99	www	
/bin/prog/shinsei	100	c:\prog\cgi\shin.exe	0x5F1F66AD	www	

1800: 通信制御ポリシー

【図20】

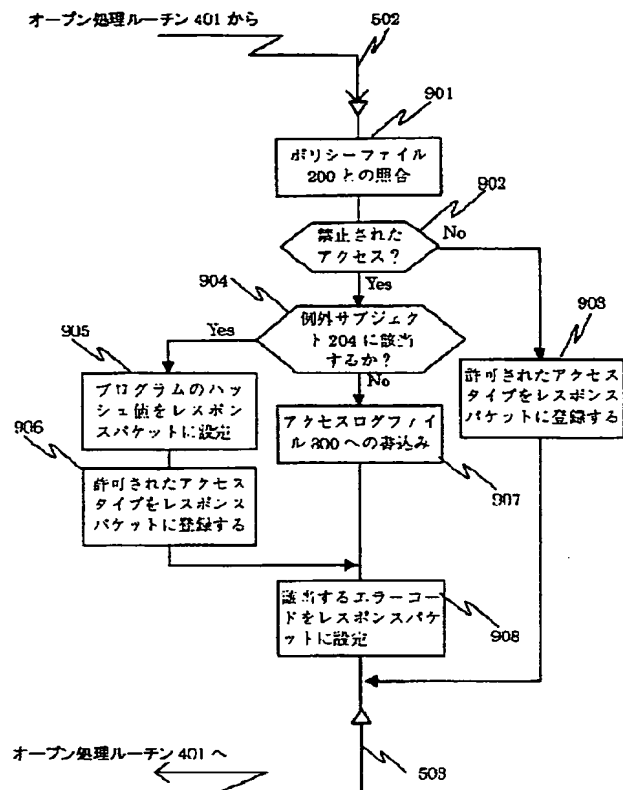
図20

2000	2001	2002	2003	2004
通信プログラムの識別子	サービス番号	暗証データ	通信中プログラムの識別子	
/bin/prog/login	30	0x228B	c:\prog\cgi\login.exe	
/bin/prog/zandaka	80	0x9A0C		
/bin/prog/shinsei	100	0x3175		

2000: 通信管理テーブル

【図 9】

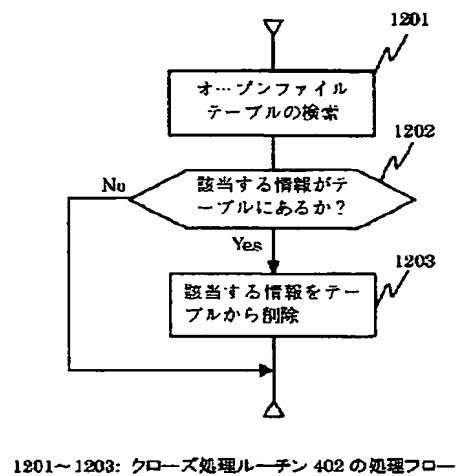
図 9



901～908: オープン制御ルーチン 405 の処理フロー

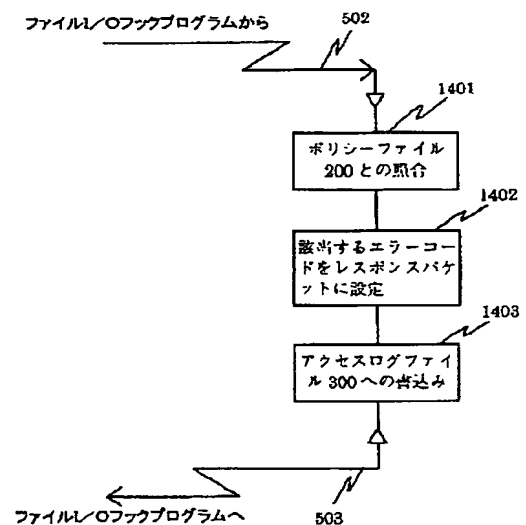
【図 12】

図 12



【図 14】

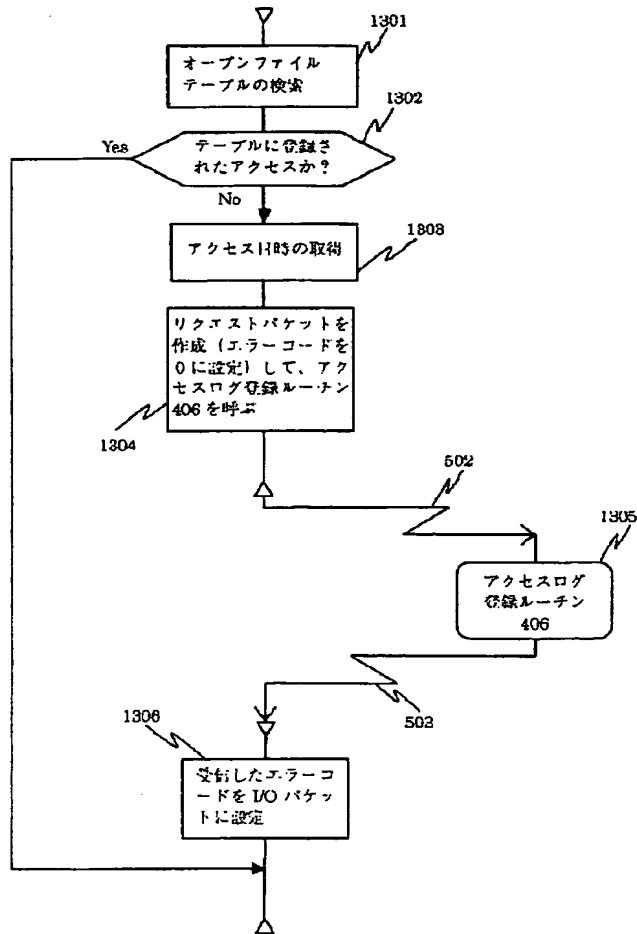
図 14



1401～1403: アクセスログ登録ルーチン 406 の処理フロー

【図13】

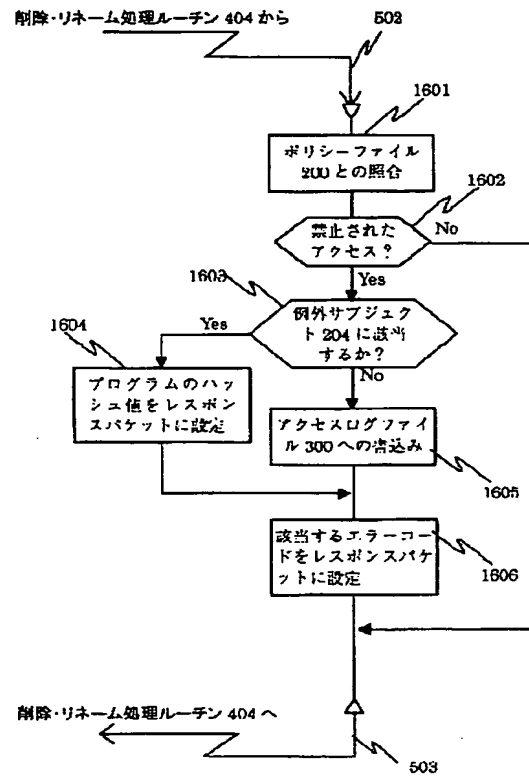
図13



1301~1306: リード・ライト処理ルーチン 403 の処理フロー

【図16】

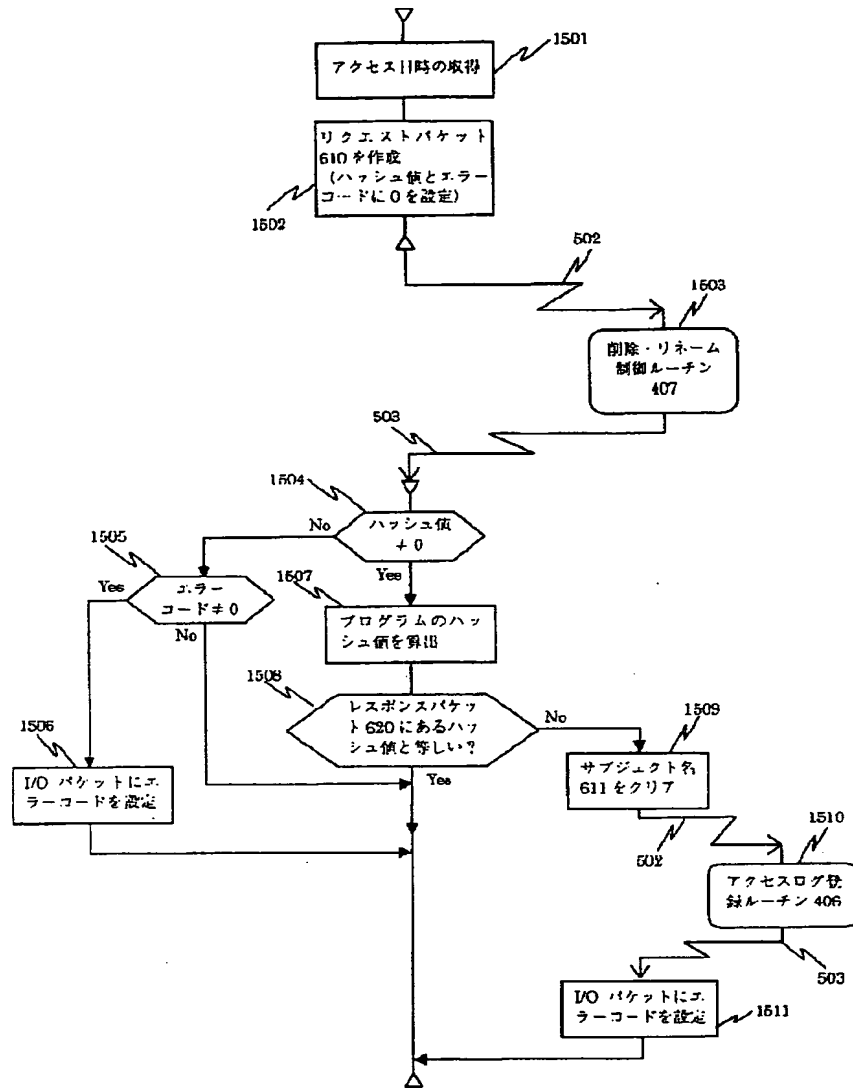
図16



1601~1606: 削除・リネーム制御ルーチン 407 の処理シーケンス

【図15】

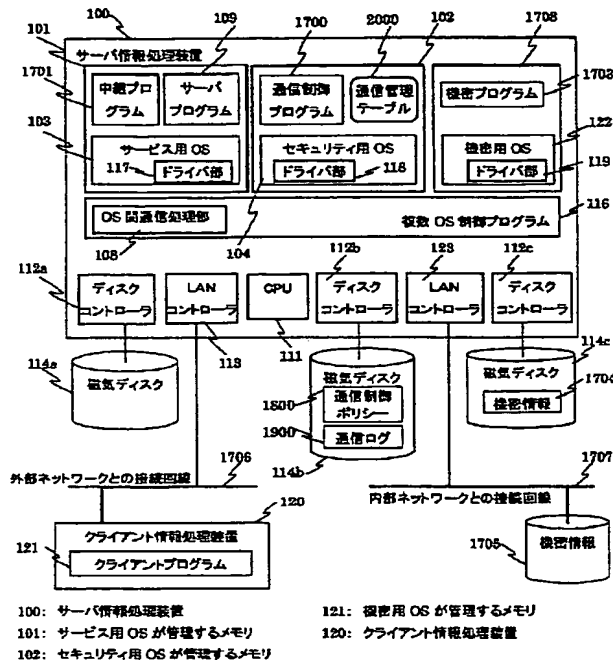
図15



1501~1509: 削除・リネーム処理ルーチン 404 の処理フロー

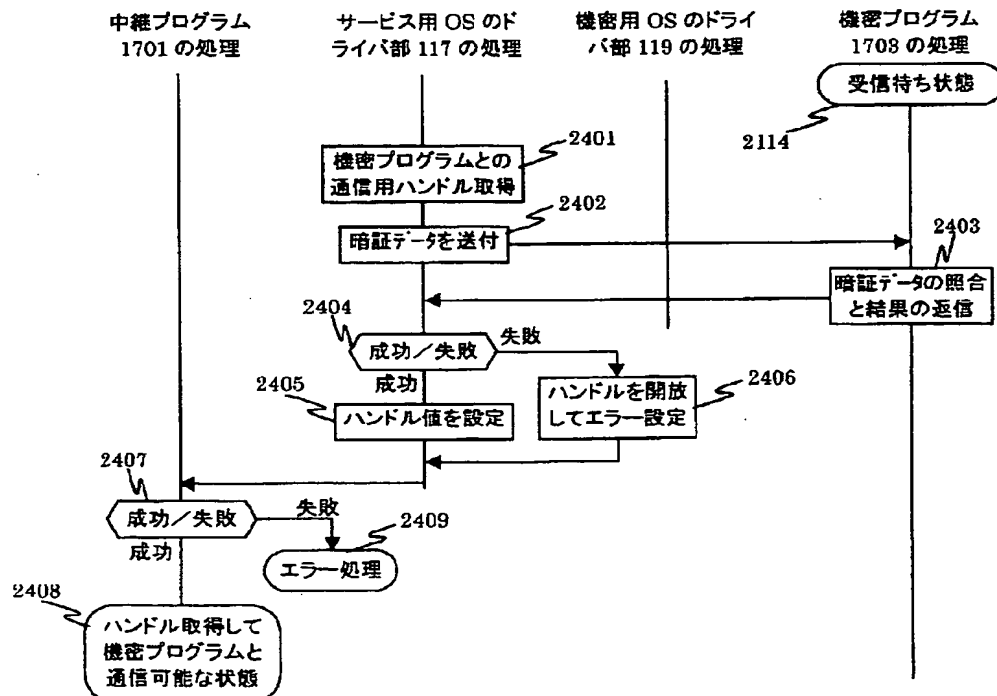
【図 17】

図 17



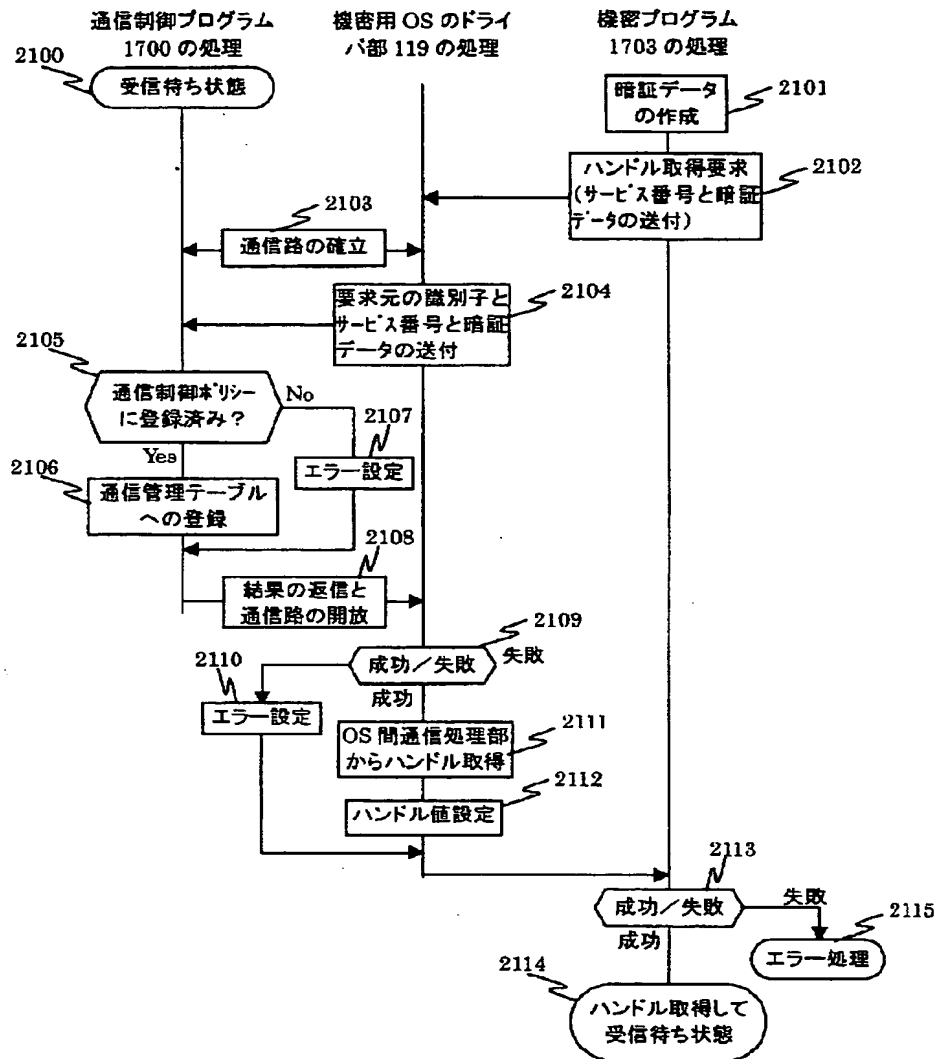
【図 24】

図 24



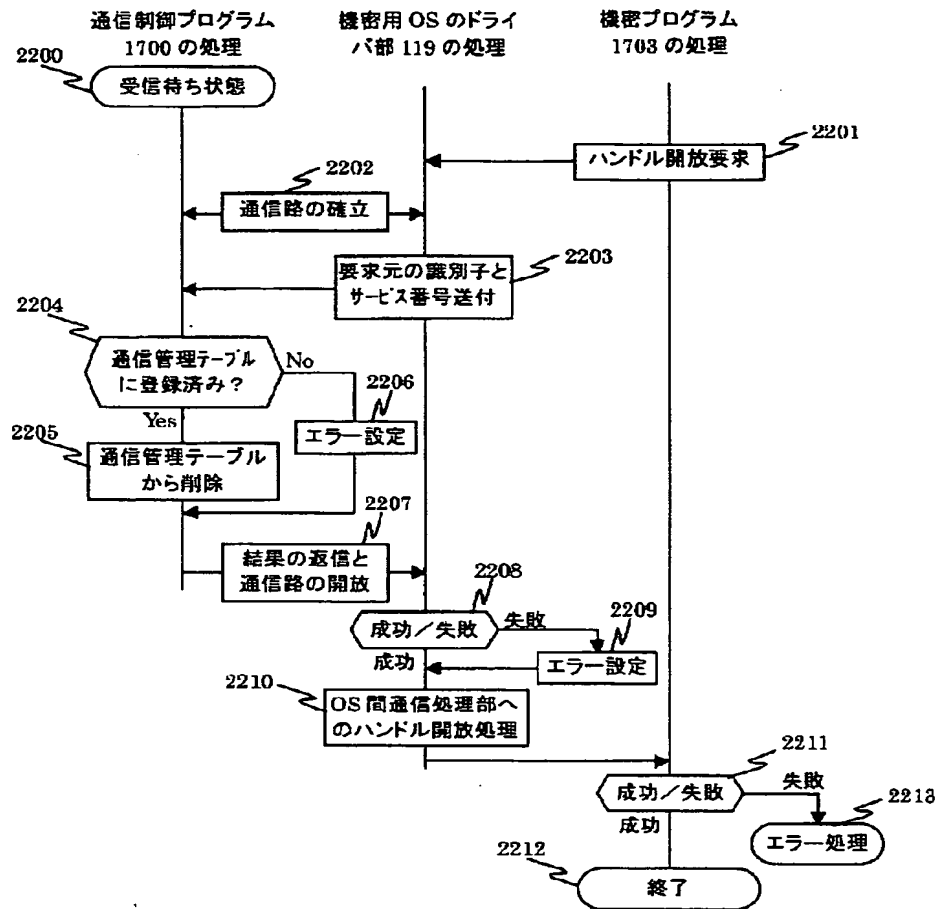
【図21】

図21



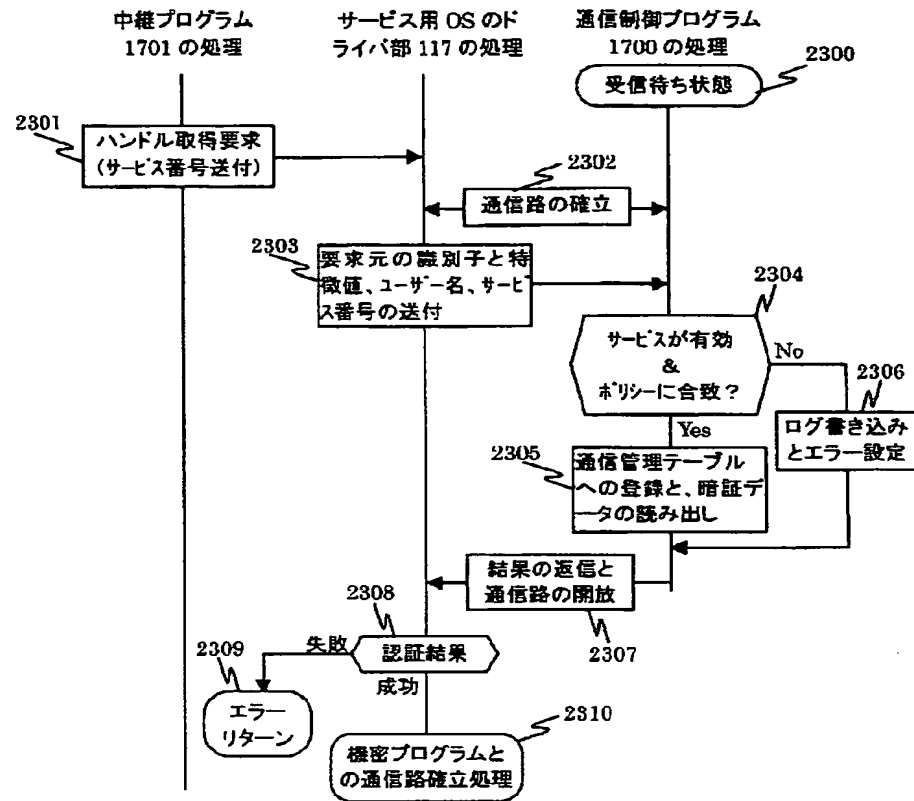
【図 22】

図 22



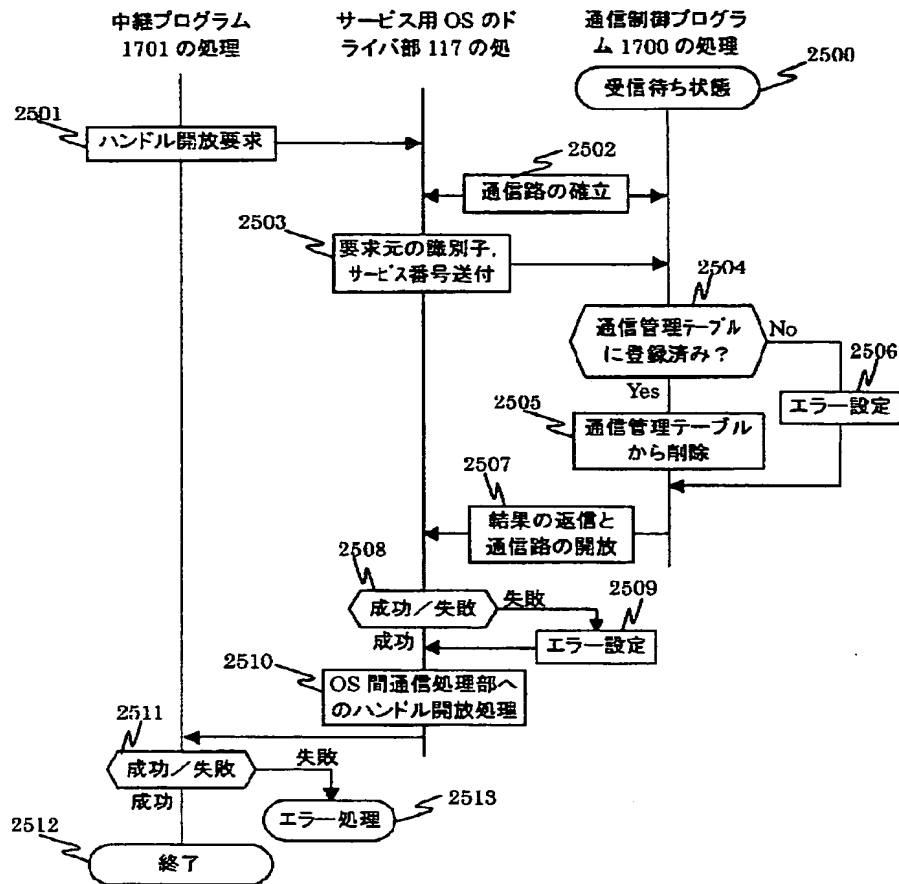
【図23】

図23



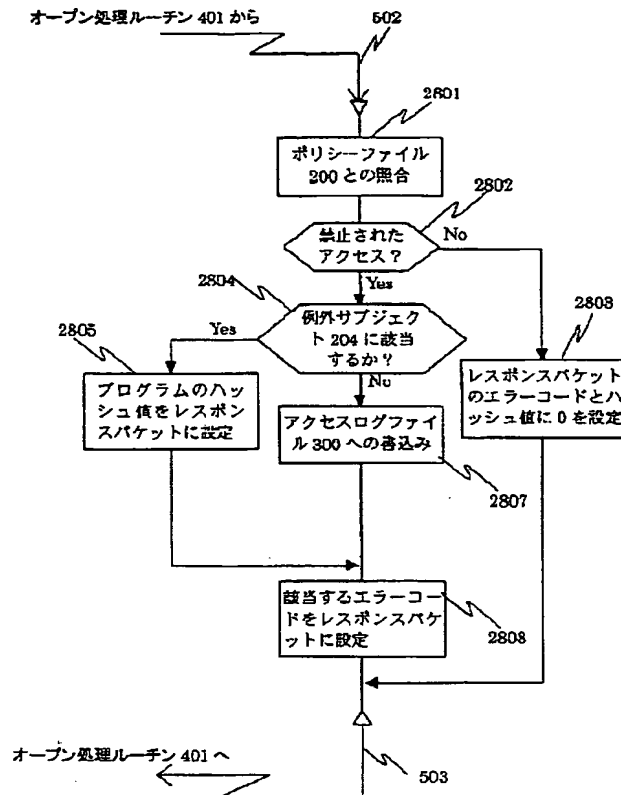
【図25】

図25



【図26】

図 26



2801～2808: オープン制御ルーチン 405 の処理フロー

フロントページの続き

(72) 発明者 梶浦 敏範
 愛知県尾張旭市晴丘町池上 1 番地 株式会
 社日立製作所情報機器事業部内

F ターム(参考) 5B017 AA01 BA06 BB06 CA16
 5B076 FB05
 5B082 GA11
 5B085 AE00 BG03 BG07